FREQUENCY PLANNING FOR COMMUNICATION NETWORKS — A GRAPH - THEORETIC APPROACH

By

N. SATYANARAYANA MURTHY

DEPARTMENT OF INDUSTRIAL AND MANAGEMENT ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

APRIL, 1991

Frequency Planning for Communication Networks —
A Graph-theoretic Approach

A Thesis Submitted

in Partial Fulfilment of the Requirements

for the Degree of

Master of Technology

by

N Satyanarayana Murthy

to the

DEPARTMENT OF INDUSTRIAL AND MANAGEMENT ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY KANPUR

April, 1991

CERTIFICATE

It is certified that the work contained in the thesis entitled "*Frequency Planning for Communication Networks — A Graph-theoretic Approach*", by N Satyanarayana Murthy, has been carried out under my supervision and that this work has not been submitted elsewhere for a degree.

A. K. Mittal
Professor
Industrial and Management Engineering Programme
I.I.T. Kanpur

April, 1991

ABSTRACT

The *Frequency Planning Problem* for large *communication networks* (like Indian Railways Microwave network) is considered where the interference-free operation of the network is managed by a one-time assignment of discrete frequencies to the various links. The problem can be formulated as a *nonlinear integer program*. However, the nonlinear integer program is difficult to solve for a reasonable size problem. Alternatively, the problem can be modelled as a *Graph Coloring Problem* with some additional constraints due to *adjacent-channel* and *intermodulation interference* mechanisms. This approach was found feasible and is followed. The graph being reasonably large and the Graph Coloring Problem being NP-Complete, some *heuristic algorithms* are sought to solve this problem.

Sequential algorithms for Graph Coloring are considered, with modifications to take care of the additional constraints due to adjacent-channel interference. A scheme of *partitioning* the graph into smaller sub-graphs, which can individually be colored quite easily and efficiently, is suggested, since interference mechanisms have a local effect. Next, the individually colored sub-graphs are put back together following a systematic method of reconciling the conflicting colors to get back the original graph. The frequency assignment corresponding to the graph coloring is evaluated for inter-modulation interference. Critical cases of interference are identified link-wise and it is left to the user to modify the frequency assignments iteratively and eliminate these cases.

ACKNOWLEDGEMENTS

Under the inspiring guidance of my thesis supervisor, Prof. Mittal, it had been a period of continuous learning for me, for nearly 8 months or so while I was carrying out my thesis work. Despite his very busy schedule, Prof. Mittal took time off and helped me keep my work running smoothly. I will always be grateful to him for all the help and advice I received from him.

All my teachers have, in their own way, helped me to build a foundation on which I was able to work further for my thesis and which also gives me hope and courage to do further work in academics. I am thankful to all of them.

My friends (and classmates) over the past two years have been a source of great joy to me. They have provided good companionship, all of them, and the kind of camaraderie all of us shared was very remarkable. The past year, when all of us were doing our thesis work, was a period of good intellectual stimulation for me and I enjoyed very much the indulgence others allowed me in their work. I must record here, my sense of gratitude towards each one of them, for all the support they gave me at various times.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

In this thesis, an attempt shall be made to provide a methodology for *frequency planning for large communication networks*, keeping in view the requirements of a typical network such as the Indian Railways Microwave communication network. In order to understand the nature of the problem, a brief introduction is given in this chapter, to the subjects of communication networks, frequency planning and interference-caused constraints on frequency planning, followed by a survey of the literature dealing with frequency planning in particular. Next, the frequency planning problem is explained in brief. An overview of the thesis and its organization is provided in the final section.

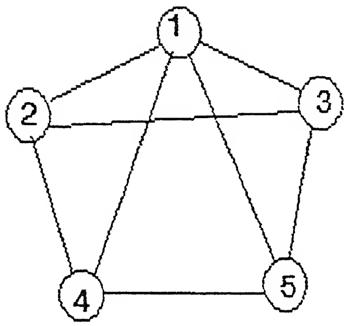
1.1 Communication Networks and Frequency Planning

In today's era of high technology, electronic communication technology is seen at the forefront. Communication networks span entire nations and most of the world, relaying a broad spectrum of signals such as data, voice and picture from one place to another. The single most precious resource used up by these networks is, perhaps, the electro-magnetic spectrum; its useful range being determined by the technology put to use.

For example, microwave networks will have a useful spectrum range from 3 GHz to 30 GHz, while domestic radio broadcasts use a much lower range of frequencies. Frequency planning is directed towards assigning different segments of the useful range of electro-magnetic spectrum to different links of the communication network with the objective of ensuring a reliable, interference-free performance of the network. Frequency planning is therefore a vital exercise towards the efficient utilization of electro-magnetic spectrum by a communication network.

A communication network can, very simplistically, be described as an undirected graph G(N,A) where N is the set of nodes which are nothing but the sites for transceivers (transmitter-cum-receiver) and A is the set of arcs joining two nodes in N, if there is a direct communication link between the two nodes. (See Figure 1.1) The actual operation of a link in a communication network is described in Appendix I.

In actual practice, the sites may be the metropolitan cities of India and the links may be the satellite links relaying telephone and television signals between the cities. A more elaborate instance of a communication network, frequency planning of which provides a background to this thesis, is that of the Indian Railways (IR), which uses the microwave (MW) range of the electro-magnetic spectrum. The frequency planning problem for such a network is explained later in this chapter (Section 4).



N = set of sites = $\{1,2,3,4,5\}$

A = set of links = $\{(1,2),(1,3),(1,4),(1,5),(2,3),$
$(2,4),(3,5),(4,5)\}$

FIGURE 1.1 Representation of a communication network by a graph G

Frequency planning for such a network implies, firstly, the creation and maintenance of a database of the transceiver sites and the communication links including all their relevant characteristics such as geographical location, transmitter power, and receiver gain. Secondly, frequency channels are assigned to links such that compatibility in terms of electro-magnetic interference (EMI) is likely to be satisfied. Next, calculations are made for the compatibility of each link, in terms of EMI, with the already planned network. If compatibility is not satisfied for a link then the frequency assignment to this link is modified. When compatibility gets satisfied, the frequency assignment to this link is documented and the planned network is updated. An extended overview of the frequency planning activity in Indian Railways, for their MW network, is given in Appendix II.

In addition to the above, frequency plans are subject to obeying the international regulations on the use of the electro-magnetic spectrum such as those implemented under the International Telecommunication Union (ITU) or CCIR. Moreover, at the national level, co-ordination is required between the various users using the same range of the spectrum.

1.2 Constraints on Frequency Planning due to Interference Mechanisms

Re-stating the objective of a frequency plan, it is to ensure that EMI is kept within tolerable limits for the operation of each link of the communication network. In this context, it will be useful to identify the main sources of EMI.

According to the classifications described, for example, by Zoellner and Beall [1], constraints imposed by EMI in the frequency plan of any communication network may be classified into three groups; namely,

(i) *Co-channel Constraints* :- Certain pairs of transmitting links cannot be assigned the same frequencies due to the locations of antennae, power of signals, and other characteristics.

(ii) *Adjacent-channel Constraints* :- Certain pairs of transmitting links should be assigned frequencies that are separated by a minimum amount in order to avoid interference.

(iii) *Co-site Constraints* :- Particular subsets of transmission frequencies should not be assigned to communication links which are located in close proximity in order to avoid potential interference such

as intermodulation (explained in Chapter II).

Appendix I provides a short introduction to the above interference mechanisms which give rise to the above constraints, how they arise and how their magnitude can be estimated by various formulae.

Technological means exist to reduce the undesirable EMI in both transmission as well as reception. Band pass filters do not allow signals outside of a specified frequency band to pass through. Antennae are of directed-type so that signal-spread is less in other directions. Polarization of signal waves is also a useful option to reduce the interference effects. Some of the above techniques receive a greater attention in Appendix I.

Given a hardware setup for operating the communication network, including all the technological means quoted above, the interference levels depend upon two factors :-

(i) *Geographical Location of the Links* :- If an antenna of one link "sees" the antenna of another link and the length of the signal path between the two antennae is significantly low then the two links can potentially interfere with each other's operation.

(ii) *The Frequency Plan for the Network* :- If two links are located such that they can potentially interfere with

each other then larger the separation between the frequencies of the two links, lesser is the actual amount of interference. So a good frequency plan takes care in assigning frequency channels to such links.

There may be other spurious sources of interference in the environment and the measure of the actual EMI may be statistics like the average number of interference cases per month, mean time of interference per month or any other suitable measure for the application at hand. In general, the interference function will be highly nonlinear both in the time domain as well as the frequency domain.

1.3 Literature Survey in the area of Frequency Planning

Mathematically, co-channel and adjacent-channel constraints are relatively easy to describe in comparision to co-site constraints. Frequency planning scenarios with these constraints have drawn some attention. Several approaches have been proposed for these type of problem scenarios.

Freeman [2], in a *Dynamic Programming* formulation, suggested building the optimum frequency plan in a stage wise manner by minimizing a certain *Interference Potential Index*. Some of the indices he suggests are *Geometric Mean Index* (geometric mean of Interference-to-Noise ratios over all links), *Mean Interference Power Index* (simple arithmetic average),

Maximum Interference Power Index (maximum over all the links) and *Interference Cases Index* (number of interference cases in the environment). The notion of *Expected Value Index* of interference is also suggested for a more rational approach.

A *Mutual Interference Chart* method due to Sachs [3] gives an analytical concept for doing compatibility calculations before making frequency assignments.

Zoellner [4] has proposed a *Simulation Game Theoretic* approach to evolve strategies for frequency assignment. A *Multivariate Search* approach to frequency assignment using *Nonlinear Programming* has been given by Massaro [5]. Cameron [6] has treated the problem as a *Graph Coloring Problem* as well as a *Set Covering Problem*.

Thuve [7] has also recast the frequency planning problem as a *Set Partitioning Problem* with the objective of minimizing the *Total System Interference*. He has however used a *Link Interference Function* which is a combined measure of the transmitter and receiver interference at the link terminals. This function is, in general, spiky and highly nonlinear,but the solution methodology is independent of it's nature. The co-channel and the adjacent-channel constraints are handled as frequency separation constraints.

More recently, the co-site constraints arising due to intermodulation interference have received some attention in the frequency planning problem. Mathur et al [8] derive an *Integer Nonlinear Programming* model to analyze the extent of intermodulation interference at a site in a communication network. They also extend the algorithms developed using this model to generate Intermodulation-Free subsets of a given set of frequencies which is useful in making compatible frequency assignments. Based on the same model, Mathur et al [9] have also described an Interactive Computer Software System for the frequency planning problem.

Box [10] describes a heuristic frequency assignment technique in which the links that prove themselves to be "difficult" in respect of interference-free assignment, rise rapidly towards the top of the list of links. The iterative method can be easily automated. However, for cases where no frequency assignment is possible with the existing resources, this method may not terminate quickly.

A survey of the applications of quantitative techniques in Radio Communication Interference problems is provided by Morito et al [11].

In a related area, Hafez et al [12] examine the effect of user's geographical distribution on the utilization of radio spectrum. In general, the remaining spectrum available for

further assignments at any time is usually less than the unused portion of the spectrum. Using appropriate measures of Spectrum Utilization Efficiency, frequency plans and strategies for frequency assignments can be evaluated for their efficacy. In this context, the authors present a new measure of true spectrum utilization and develop a model to calculate it.

1.4 The Frequency Planning Problem for Indian Railways Microwave Communication Network

The IR-MW network has about 100 sites and 300 links spanning the entire Indian nation. A brief description of frequency planning activity for Indian Railways MW network is given in Appendix II. Each link in the network operates on a discrete frequency channel. As the number of frequency channels available for use is limited, they have to be shared between the large number of links. A signal transmitted along a link may stray into the receivers of a neighbouring link and cause interference problems there. The magnitude of interference is significant whenever both the links are operating on the same frequency channel or adjacent frequency channels. The frequency planning problem (FPP) is to allocate the frequency channels to the links such that there are no significant interference problems for any of the links.

1.5 Overview of the thesis

The thesis is organized in the following chapters :-

Chapter II : describes a mathematical model for predicting interference magnitudes and it's extension to a nonlinear integer program formulation of FPP. The FPP is also formulated as a graph coloring problem. The nonlinear integer program formulation for evaluating the magnitude of a special type of interference problem, the intermodulation interference, is presented.

Chapter III : Solution methodologies are developed based on the graph coloring algorithms, of which the sequential algorithms are reviewed in detail. A graph partitioning procedure is examined next, in the context of following a divide-and-conquer strategy of coloring the large graphs encountered in the problem. A heuristic method of merging the colored partitions without violating the coloring constraints is presented. Finally, a dual backtrack algorithm for solving the nonlinear integer program formulation of the intermodulation interference evaluation problem is discussed.

Chapter IV : The implementation of computer programs to solve the FPP, using the solution methodologies of Chapter III, is described. A solved example using the implemented system is presented for a problem segment taken from the Indian Railways

Microwave network.

Appendix I : The technical details of the operation of the network is explained in a simplified manner. The propagation formulae for interference calculations are also given. The mechanisms by which the interference problems arise are explained at the end.

Appendix II : The frequency planning activities currently going on in the Indian Railways for their microwave communication network are described.

CHAPTER II

MATHEMATICAL DESCRIPTION OF THE FREQUENCY PLANNING PROBLEM

In this chapter, the mathematical model of interference is ·esented first, which also draws upon the real-life data ·ovided by "The Approach Papers for Frequency Planning for ıdian Railways Microwave Network" [13]. Next, two alternative ıthematical formulations are described for the Frequency :anning Problem (FPP), namely, the Nonlinear 0-1 Integer Program ›del and the Graph Coloring Problem model. Each formulation is :companied by a literature survey of its domain. Finally, an ıportant type of co-site constraint - the intermodulation ıterference - is discussed in mathematical terms, based on the terature published in this area.

1 A Mathematical model for Interference

Consider a communication network as described below :-

i = an index which uniquely identifies a link of the network, where $1 \leq i \leq n$;

F_i = frequency assigned to link i, where F_i takes discrete values from a given set F of frequencies;

U_i = upper limit for interference level in link i, in the same units in which interference is measured;

CA_i = set of neighbouring links of link i within the limits of co-ordination area of link i,
where co-ordination area of a link is the area around it such that other links within this area can potentially interfere with its operation;

f_i = a mathematical function which expresses the interference level in link i due to interference from various links in CA_i (in the same units in which interference is measured);

The constraint on interference level in each link of the network is expressed as

$$f_i(F_i, F_j \ldots F_k) \leq U_i \quad \forall\ i;$$
$$\text{where } F_j \ldots F_k \in CA_i \tag{2.1}$$

In general, the expression for f_i is not available in an explicit mathematical form. As per "The Approach Papers for Frequency Planning for Indian Railways Microwave Network" [13], the expression for interference between two links i and j, f_{ij}, contains two distinct and additive terms, a_{RFij} and IRF_{ij} ,where a_{RFij} is the RF-decoupling between the two links and IRF_{ij} is the Interference Reduction Factor which depends upon the frequencies F_i and F_j assigned to the links i and j respectively.

2.1.1 RF-decoupling

RF-decoupling between two links indicates the extent to which operation of one link is free from interference due to operation of the other link. Its magnitude a_{RF} can be expressed as

$$a_{RF} = D + a_T + a_R + \Delta G + \Delta a + \delta ,$$

where

$D = 20 \log \left[\frac{D_{is}}{D_{ws}}\right]$ = difference between propagation loss in interfering and wanted signal paths;

a_T = angle attenuation at interfering transmitter;

a_R = angle attenuation at interfered receiver;

ΔG = difference between antenna gains at wanted and interfering signal transmitters;

Δa = difference between attenuations at wanted and interfering signal transmitters;

δ = additional obstructions in interfering signal path.

Higher the value of a_{RF} is, for two links, larger is the extent to which the two links are free from mutual interference. A couple of points need to be noted about a_{RF}.

(i) It is dependent upon the terrestrial layout of links

and sites of the network, as also upon the hardware installed at the various sites.

(ii) It is independent of the frequency assigned to the links.

2.1.2 Interference Reduction Factor

The RF-decoupling a_{RFij}, obtained for two links i and j, assumes that links i and j have the same frequency. This is the worst possible case of interference between links i and j. By assigning different frequencies F_i and F_j, to links i and j respectively, the RF-decoupling is further increased by the amount IRF_{ij} and therefore, interference is reduced. That is why the term IRF is called Interference Reduction Factor.

There is no single or simple expression for calculating IRF_{ij} for any two links i and j. Curves (graphs) are available which depict IRF_{ij} as a nonlinear function of ΔF_{ij}, where

$$\Delta F_{ij} = |F_i - F_j| .$$

In contrast with a_{RFij}, it is to be noted that IRF_{ij} depends upon the frequencies allocated to links i and j. Even the graphical curve for IRF_{ij} is not unique, the curve to be used depends upon the fixed characteristics of the radio relay systems used by the links i and j (FM 120, FM 300, FM 960 and DRS 34).

Since there is no easy expression for IRF_{ij} calculations, following schemes may be considered :-

(i) *Tabulation* : Tables will be created for showing the variation of IRF_{ij} with ΔF_{ij}. Based on the fixed characteristics of the radio relay systems used by links i and j, a particular table may be selected and using $\Delta F_{ij} = |F_i - F_j|$, the value of IRF_{ij} may be read out.

(ii) *Piece-wise Linear Approximation of IRF_{ij} curves* : The piece-wise linear functions developed as an approximation to the original IRF curves may be used in their mathematical form to calculate IRF at any given ΔF.

2.1.3 The Basic Interference Equation

So the basic mathematical model of interference (2.1) now gets modified as in Equation (2.2) which follows.

$$\sum_{\substack{j \in CA_i \\ i \neq j}} (a_{\mathbf{RF}\,ij} + IRF_{ij}) \geq \Delta_i \qquad \forall\, i \tag{2.2}$$

where

Δ_i = minimum level of RF-decoupling required for operation of link i, and

$IRF_{ij} = f_{ij}(|F_i - F_j|)$, and

$a_{RF\ ij}$ is independent of F_i and F_j.

2.2 A Nonlinear Integer Program formulation of FPP

The basic nature of the Frequency Planning Problem (FPP) is allocation from a set F of frequencies, exactly one frequency k to each link i in the communication network. To formulate this problem , the decision variables chosen are assignment variables of 0-1 integer type, *i.e.*,

Decision Variable :

$$X_{ik} = 1, \text{ if frequency } k \text{ is assigned to link } i;$$
$$= 0, \text{ else.}$$

Constraints :

$$C1 :- \sum_{k \in F} X_{ik} = 1, \quad \forall\, i, \quad \text{where } 1 \le i \le n. \qquad (2.3)$$

The above constraint implies that each link i is assigned exactly one frequency k from the set F.

$$C2 :- D(i) \ge d_i, \quad \forall\, i, \quad \text{where } 1 \le i \le n; \qquad (2.4)$$

$D(i)$ = an expression for RF-decoupling of link i from the rest of the links in the network as a result of a particular frequency allocation; and

d_i = minimum level of RF-decoupling required for link i.

2.2.1 Expression for D(i)

Given a table of IRF values which can be referred by the values IRF_{ij} for any two links i and j having frequencies F_i and F_j respectively, or alternatively, given a function IRF_{ij} to calculate the IRF values, where

$$IRF_{ij} \quad = \quad f_{ij} \left(|F_i - F_j| \right),$$

$D(i)$ can be evaluated in the manner described below.

Assume frequency k is assigned to link i, and frequency l is assigned to link j, *i.e.*,

$$X_{ik} = 1, \quad \text{and} \quad X_{jl} = 1.$$
$$(F_i = k) \qquad\qquad (F_j = l)$$

Then,

$$\sum_{j \in CA_i} \sum_{l \in F} IRF_{ij} \, X_{jl} = D(i), \quad \text{if} \quad \underset{(F_i = k)}{X_{ik} = 1.} \tag{2.5}$$

However, k is unknown *apriori*; therefore

$$D(i) \quad = \quad \sum_{k \in F} \Big(\sum_{j \in CA_i} \sum_{l \in F} IRF_{ij} \, X_{jl} \Big) X_{ik} \, . \tag{2.6}$$

This expression for $D(i)$ is to be read in conjunction with constraint C1 so that summation over k (or l) belonging to F, makes sense.

Some points are to be noted in the expression for $D(i)$ as derived above :-

(i) It was assumed, by summing over links j in CA_i, that there is negligible interference from all links which are not in CA_i.

(ii) By summing over all links j in CA_i, it was also assumed that RF-decoupling of link i from rest of the links in the network is the sum of RF-decoupling values of link i from each link j in CA_i.

There is another possibility of formulating the constraint C2 for every pair of links (i,j) where j is in CA_i, by considering individually the RF-decoupling of link i with each link j in CA_i, instead of performing a sum over them as has been done above.

(iii) The expression for $D(i)$ is nonlinear as it contains product terms like $(X_{ik}.X_{jl})$.

2.2.2 The Nonlinear Integer Program for FPP

The nonlinear integer program formulation in it's final form is as given below.

$$C1 :- \sum_{k \in F} X_{ik} = 1, \quad \forall\, i, \quad \text{where} \quad 1 \leq i \leq n. \tag{2.7}$$

$$C2 :- \sum_{k \in F} \sum_{j \in CA_i} \sum_{l \in F} IRF_{ij}\, X_{ik}\, X_{jl} \geq d_i, \quad \forall\, i, \tag{2.8}$$

where $1 \leq i \leq n$, and

$X_{ik} = 0$ or 1.

It is noted that there is no objective function in the above formulation. The objective function is trivial in this case because any feasible solution to the above problem is sufficient to provide a frequency plan for the network. Broadly speaking, the objective is to conserve the spectrum resources *i.e.* to use minimum number of frequencies in preparing a plan. The intermodulation phenomenon (Refer Appendix I for details) has not been considered in the above problem formulation, so the frequency plan obtained by solving the above problem is subject to verification for the presence or absence of intermodulation interference. The problem of verification for the presence or absence of intermodulation interference at any site in the network is formulated in a later section of this chapter.

It is also noted that the problem size is very large for a communication network akin to the Indian Railways Microwave network. For allocating about 60 channels, available from 4 relay systems in use, to approximately 300 active links there will be 60 x 300 = 18000 decision variables X_{ik} of 0-1 integer type. The number of nonlinear constraints will be equal to the number of links *i.e.* about 300. Linearisation of the quadratic terms in equation (2.8), as a part of the solution strategy for solving the nonlinear program, will further multiply the number

of variables as well as the number of constraints.

2.2.3 Literature Survey

There are many approaches to solve a nonlinear integer program. Hansen [14] has reviewed the various methods of nonlinear 0-1 integer programming based on an extensive survey of existing literature. Three main approaches have been : enumerative methods, boolean algebraic methods and reduction to equivalent problems. Enumerative methods are basically branch and bound schemes of solving the 0-1 nonlinear program directly. Reduction to equivalent problems involve converting the problem to a polynomial 0-1 program for which many algorithms have been devised. One of the most important techniques is linearization in which the polynomial 0-1 program is reduced to a linear 0-1 program by replacing all product terms $\prod_{i \in J} x_i$ by a new variable z. Two new constraints get added :

$$-\sum_{i \in J} x_i + z + |J| - 1 \geq 0, \qquad \text{and}$$

$$-\sum_{i \in J} x_i - |J|\, z \geq 0.$$

Even small nonlinear 0-1 programs can get enlarged considerably by this method. Glover and Woolsey [15] have proposed several ways of economizing on the number of new constraints introduced in the process of linearization. A more recent survey of the various general algorithms developed for a nonlinear integer program appears in Cooper [16].

2.3 A Graph Coloring Problem Formulation of FPP

A Graph Coloring Problem formulation of the Frequency Planning Problem can be obtained by viewing the communication network as a graph G(N,A), where

N = set of sites in the communication network; and

A = set of links where each link joins two sites in N uniquely.

Based on the geographical location of the links (and sites) and the communications hardware assemblage at the sites *viz.* antenna, transmitter/receiver, bandpass filters etc., link-to-link interference potential (IP) is calculated for every pair of links within co-ordination distance of each other assuming that all links are operating at the same frequency. These values are stored in a matrix IP,

$IP = (IP_{ij})$, where $(i,j) \in A$, and $j \in CA_i$ or $i \in CA_j$.

Stipulating a tolerance level of this interference potential for each link i (based on the number of links in CA_i), a secondary graph GC is developed in the following manner :-

(i) Each link i in the graph G becomes a vertex i in GC,

(ii) Two vertices i and j in GC are joined by an edge if the two links i and j they represent have a mutual interference potential greater than the stipulated tolerance level for either link i or link j.

The secondary graph GC indicates all the critical interference linkages between the links of the communication network. The criticality here is defined with respect tp the stipulated level of tolerance of the mutual interference potential between two links.

The objective of frequency planning is to allocate a discrete frequency to each vertex of GC which is a link of the communication network. If two vertices which are joined by an edge in GC are assigned the same frequency then the interference between the two links they represent will obviously be unacceptably high because it exceeds the stipulated tolerance level of interference (This is nothing but the Co-channel constraint described earlier in Chapter I and explained in in greater detail in Appendix I). However if they are assigned different frequencies then the interference decreases sharply and there is no interference problem of the original order of magnitude. Since the frequencies which are allocated are drawn from a limited set F, it makes sense to minimise the actual number of frequencies used in the allocation process. It might also be desirable to conserve the scarcely available frequencies for fulfilling the future requirements of the communication network.

By treating the discrete frequencies as colors, it is now possible to visualise the FPP as a Graph Coloring Problem on

the graph GC. In a Graph Coloring Problem, it is sought to color all the vertices of a graph using minimum number of colors such that no two adjacent vertices (vertices which are joined by an edge) have the same color.

2.3.1 Additional constraints on Graph Coloring based on Adjacent channel constraints in Frequency Planning

Sometimes, even if two neighbouring vertices in GC are assigned adjacent frequencies (adjacent in the frequency spectrum) there might be interference problems (This is how Adjacent-channel constraints arise). So the frequency allocation constraint is modified in the sense that two neighbouring vertices in GC cannot be assigned frequencies that are the same or adjacent to each other *i.e.*, they must have a separation of at least one frequency channel. For the Graph Coloring Problem, it is equivalent to saying that all the vertices of the graph GC are to be colored using minimum number of colors such that no two neighbouring vertices are assigned the same or adjacent colors. Two colors are adjacent if the frequencies they represent are adjacent in the spectrum.

2.3.2 Some characteristics of the formulation

Following points may be noted in the formulation described above :-

(i) The above process of tightening the frequency allocation constraint may be extended further. For example, the frequency allocation can be constrained such that two adjacent vertices in GC cannot be assigned frequencies that are separated by less than 2 (or 3 or 4 or ...) frequency channels.

(ii) Once again, intermodulation interference is not considered in the above formulation. It has to be checked separately after the frequency allocation is specified completely. As stated earlier, the formulation of the problem of evaluating intermodulation interference is taken up under Section 4 of this chapter.

2.3.3 Literature Survey of the Graph Coloring Problem

The Graph Coloring Problem has been widely researched, mainly on account of the famous Four Color Problem (FCP). The Four Color Problem is the problem of coloring a planar graph using at most 4 colors. This problem was posed more than a hundred years ago by cartographers (map-makers) who found by experience that they could color any map, in which any two countries sharing a common border are colored differently, with a maximum of 4 colors (A geographical map can be represented by a planar graph). But no proof of this problem was forthcoming until a decade back when a computer was used to analyse around

thousand special cases of the FCP to show conclusively that the FCP had indeed a proof and a solution. This solution is due to K. Appel, W. Haken, and J.A. Koch [17]. It may be noted that the graph GC, as derived from the communication network, is rarely planar in nature.

In light of the fact that the FCP is a special case of the Graph Coloring Problem, and has itself defied a solution for close to 150 years - being solved ultimately only with the aid of a computer, it would seem likely that there is something inherently difficult about the Graph Coloring Problem. Indeed with the development of the theory of computational complexity of combinatorial problems, mainly by Garey and Johnson [18], it has been found that the Graph Coloring Problem, also known as the Chromatic Number Problem, is NP-Complete ([19], [20]). Therefore, it seems unlikely that any polynomial time-bounded algorithm exists for solving this problem optimally.

Graph coloring has considerable applications in the areas of combinatorial optimization. Problems involving conflict resolution or optimal partitioning of mutually exclusive events can be modelled as graph coloring problems. The examination scheduling problem, in which examinations are to be scheduled in the smallest number of time periods such that no individual is required to participate in two examinations simultaneously, is modelled as a graph coloring problem with examinations as vertices of a graph and time periods as colors. Two vertices are

joined by an edge if both the corresponding examinations require the participation of an individual. So the problem of coloring the graph using minimum number of colors is equivalent to using minimum number of time periods to schedule the examinations.

Storage of chemicals on the minimum number of shelves such that, no two mutually dangerous (reactive) chemicals are stored on the same shelf, can also be modelled as a graph coloring problem. Machine and job scheduling and loading problems have also been defined as graph coloring problems.

The constraints in a problem, which can be modelled as a graph coloring problem, are usually expressible in the form of pairs of (in)compatible objects. Such (in)compatibilities are represented by the edges of the graph where the objects become the vertices. In practice, the constraints imposed are more complex than incompatibility constraints (Adjacent-channel constraint is one example) and hence the models developed are no longer simple coloring problems.

The methodologies known for the solution of the Graph Coloring Problem are reviewed from published literature in the first part of Chapter III.

2.4 A Mathematical Description of Intermodulation Interference

An important type of co-site constraint is the *intermodulation interference* which may be expressed mathematically unlike some other constraints on frequency planning. The intermodulation phenomenon is the creation of secondary frequencies in the transmitter/receiver circuitry as a result of interaction of multiple active frequencies at a site. These secondary frequencies can interfere with other signals and distort the quality of radio communication. Thus in a large and dense communication network, it is very crucial to minimize this type of interference even though it may be of secondary nature.

2.4.1 Mathematical condition for the occurence of Intermodulation Interference

Consider a site which is communicating directly with n other sites using n links. Let F_i be the frequency assigned to link i, where $1 \leq i \leq n$. The intermodulation interference evaluation problem is to determine if the set of frequencies F_1, F_2, ..., F_n operating simultaneously at the site is free of the intermodulation phenomenon upto a pre-specified order Q or less (Order Q is defined a little later). It turns out that the transmission frequencies create secondary frequencies, which are integer linear combinations of F_1, F_2, ..., F_n. If one of these secondary frequencies is the same as a receiving frequency, which could be one of the frequencies F_1, F_2, ..., F_n, then

intermodulation interference occurs. It should be noted that each of the frequencies F_1, F_2, ..., F_n can be used as a transmission frequency only or a receiving frequency only at a given time. Mathematically, intermodulation interference therefore occurs whenever

$$\sum_{i=1}^{n} F_i X_i = 0, \quad \text{for some integers } X_i \text{ where } 1 \le i \le n,$$

and at least one of the X_i takes the value $\pm$ 1.

The sum $Q = \sum_{i=1}^{n} |X_i|$ is defined as the order of interference. It turns out that the interference is strong if the order Q is small. Hence finding the lowest order of intermodulation is useful in evaluating a given frequency assignment, in this case, the set of frequencies F_1, F_2, ..., F_n assigned to the links 1..n incident at the site under consideration.

2.4.2 A nonlinear integer program formulation for evaluation of Intermodulation Interference

The problem of evaluating whether the frequency assignment to the links incident at a site is free of intermodulation interference has been formulated by Mathur et al [8] and can be stated as :-

$$\text{Min.} \quad Q_0 = \sum^{n} |X_i| \tag{2.9}$$

$$\text{s.t.} \qquad \sum_{i=1}^{n} F_i X_i = 0, \tag{2.10}$$

where $1 \le i \le n$, and X are integers, and

$\exists$ at least one i s.t. $|X_i| = 1$. (2.11)

If $Q_0^{opt} > Q + 1$ then the set of frequencies $F_1, F_2, \ldots, F_n$ can be declared as intermodulation-free.

In the real world, the receiving frequency is interfered with if the secondary frequency due to intermodulation is close, but not necessarily equal, to the receiving frequency. As a result the constraint (2.10) is modified so that

$$-GB \le \sum_{i=1}^{n} F_i X_i \le GB, \tag{2.12}$$

where GB is a known parameter called Guard Band.

It may be noted that the above problem has to be solved for each site in the network. Moreover it can be solved only when the frequency assignment is known *apriori* for the communication network. A solution to this problem does not generate a frequency plan, rather it evaluates the given frequency plan with respect to the phenomenon of intermodulation interference at the various sites in the network. However this will aid the process of preparing a feasible frequency plan.

2.4.3 Literature Survey of solution methodologies

From the problem formulation, it is clear that it can be reduced to a nonlinear integer program, the modulus operator in the objective function being the source of nonlinearity. While any of the general algorithms for a nonlinear integer program may be used for solving the above problem, efficient solution methods must be developed, as this problem has to be solved many number of times for evaluating a given frequency plan.

Two backtrack algorithms proposed by Morito *et al* [21] have been examined by Morito *et al* [22], one solves the primal problem while the second solves the dual. The dual backtrack algorithm appears to be computationally favourable and is discussed in greater detail in Chapter III. Nishimura *et al* [23] also investigate two branch-and-search techniques, again for primal and dual versions of the problem, for generating subsets of frequencies which are free from intermodulation.

2.5 Final Choice of the Problem Formulation

The Graph Coloring Problem formulation is taken up for developing solution methodologies for the FPP, in preference to the Nonlinear 0-1 Program approach. The latter formulation is very bulky in size and is not handled easily from computational point of view. The graph coloring approach has a compact

structure in the form of graph and it has a better defined objective of minimizing the number of colors used *i.e.* ocnserving the frequency resources. Moreover, a large variety of graph coloring algorithms are available in literature for ready use.

CHAPTER III

SOLUTION METHODOLOGY FOR THE FREQUENCY PLANNING PROBLEM

In this chapter, the solution methodologies for solving the FPP are developed in accordance with the Graph Coloring Problem formulation, described in the previous chapter. Initially the various *graph coloring routines* are explained followed by a description of the modifications necessitated in order to accomodate the Adjacent-channel constraint in frequency planning. As the graph GC, derived from the communication network, is large-sized and the interference effects are of local nature, a *divide-and-conquer strategy* is devised for *partitioning the graph* based on it's edge connectivity. After recursively coloring the partitions, a procedure is developed for *merging* them back to the original graph GC, reconciling the vertex colorings wherever any conflict arises. All the procedures mentioned above are presented along with the basic theoretical background. Additionally, reference is invited to the literature published in the associated domains. The solution methodology for carrying out the *intermodulation interference evaluation* of the frequency plan generated by the coloring of the graph GC is also discussed based on the nonlinear integer program model given under section 4 of Chapter II.

3.1 Graph Coloring Algorithms

There have been two major approaches to vertex coloring of a graph - *independent set approach* and *sequential approach*. As is to be expected, each approach has it's own advantages and drawbacks. Of course, there are also hybrid approaches which try to seek the benefits of both the approaches by combining the strategy of sequential coloring with independent set approach. Both the approaches are discussed below, under separate sections.

3.1.1 Independent Set Approach

Some definitions borrowed from Syslo *et al* [24] are necessary in order to explain the independent set approach.

[Definition] A subset of vertices of a graph in which no two vertices are joined by an edge is called an *independent set*. ■

[Definition] A subset W of vertices is a *maximal independent set* (MIS) if there is no independent set in which W is contained properly. ■

[Definition] A *maximum independent set* (MmIS), on the other hand, is a maximum cardinality independent set. ■

A coloring of the vertices of a graph G using k colors is equivalent to a partition of the set of vertices of G

into k independent sets $V_1, V_2, \ldots, V_k$, such that

$$V_i \cap V_j \quad = \quad \phi \quad \text{for } i \neq j, \text{ and } i,j = 1, \ldots, k, \text{ and}$$

$$\bigcup_{i=1}^{k} V_i \quad = \quad V.$$

Such a partition is called a k-coloring partition of V. The independent set approach is to discover such a partition by first coloring vertices with color 1 *i.e.* finding set V_1, then set V_2, and so on upto set V_k.

To understand the basic relationship, following which various algorithms, both exact and approximate, have been developed for solving the graph coloring problem, Theorem (3.1) is quoted below from Syslo *et al* [24].

[Theorem 3.1] For every k-coloring partition $(V_1, V_2, \ldots, V_k)$ of a graph G there exists a l-coloring partition $(V_1', V_2', \ldots, V_l')$ of G such that $l \leq k$ and at least one of the sets in the latter partition is a MIS.

Proof :- If V_1 is not a MIS of G then it can be augmented by the vertices of the other subsets to become such a set V_1'. Now, every set $V_i' = V_i - V_1'$ for $i = 2, 3, \ldots, k$ is independent, however, some of them may be empty. Dropping empty sets $V_i' = \phi$, if any, V_i' form a~~n~~ l-coloring partition of G for $i = 1, 2, \ldots, k$ where $l \leq k$, and V_1' is a MIS. ∎

It follows from Theorem (3.1) that in every partition

of G due to an optimal coloring, all sets are MIS. Let W be one such MIS of G. Then all vertices in W can be colored by the same color. Let $G|_{V-W}$ be the subgraph induced by the set of vertices V - W. Hence

$$\chi(G) = \min_{W \subseteq V} \chi(G|_{V-W}) + 1 \qquad (3.1)$$

where $\chi(G)$ denotes the number of colors used in an optimal coloring of G, also called the *chromatic number* of G and it is assumed that $\chi(\phi) = 0$.

Equation (3.1) is the basis for the various algorithms which follow the independent set approach.

3.1.1.1 An exact method for optimal vertex coloring of a graph

An exact procedure for coloring a graph using the independent set approach is a backtracking method, as most of the other exact methods are. If a graph G is k-chromatic, then it can be colored with k colors; coloring first with color 1 a MIS W_1 of G, next coloring with another color 2 a MIS W_2 of $G|_{V-W_1}$, and so on until all the vertices of G are colored. However, this method does not specify which MIS should be colored at each step; therfore, every MIS of a current subgraph must be considered. That is precisely the reason for backtracking in such a method. An algorithm due to Christofides [25], illustrates the above procedure.

3.1.1.2 Approximate algorithms

The idea of generating a set of vertices of one color at a time gave rise to several approximate algorithms. The maximum independent set algorithm (MmIS algorithm) generates a sequence of maximum cardinality independent subsets in subgraphs of uncolored vertices and colors them one by one.

The subproblem of finding a MmIS in a graph has no polynomial-time algorithm, so an efficient algorithm is used to generate an approximately MmIS. This becomes the genesis of the approximately maximum independent set (AMmIS) algorithm.

Both the versions of the approximate coloring algorithm can be found in Syslo *et al* [24]. Interestingly enough, while the above approximate algorithms, based on the independent set approach, exhibit better worst-case behaviour than the algorithms based on the sequential approach described below, in their average behaviour they produce substantially inferior colorings.

3.1.2 Sequential Approach

Let $v_1, v_2, \ldots, v_n$ be an ordering of the vertices of a graph G. In a sequential method for coloring the vertices of G, vertex v_i is added to the subgraph induced by already colored vertices $v_1, v_2, \ldots, v_{i-1}$ and a new coloring of $v_1, v_2, \ldots, v_{i-1}, v_i$ is determined. This step is repeated for $i = 1, 2,$

..., n, where for $i = 1$ the subgraph is empty. At each step an attempt is made to use relatively small number of colors.

The following sequential algorithms are reviewed as in a paper by Matula *et al* [26].

3.1.2.1 Simple Sequential Algorithms

A simple sequential algorithm goes as follows :-

ALGORITHM SIMPLE_SEQUENTIAL;

STEP 1 : *COLOR*(v_1) := 1;

STEP 2 : *FOR* i := 2, 3, ..., n *DO*

COLOR(v_i) := *MIN* {$(k : k \geq 1)$ *AND* *COLOR*$(v_j) \neq k$ *for every* v_j $(1 \leq j \leq i)$ *adjacent to* v_i}

END.

It is noted that in this simple algorithm the vertices colored prior to coloring v_i retain their colors. Each vertex v_i can be colored by color i, therefore COLOR$(v_i) \leq i$. On the other hand, at least one of the first $\deg(v_i) + 1$ colors can be assigned to v_i. Hence,

$$\mathrm{COLOR}(v_i) \leq \mathrm{Min}\ \{i,\ \deg(v_i) + 1\} \tag{3.2}$$

for every $i = 1, 2, ..., n$, and thus

$$\chi_S(G) \quad \leq \quad \max_{1 \leq i \leq n} \quad (\mathrm{Min}\ \{i,\ \deg(v_i) + 1\}). \tag{3.3}$$

Alternatively,

$$u_S(G;\ v_1,\ v_2,\ \ldots,\ v_n) = \max_{1 \leq i \leq n} \quad (\mathrm{Min}\ \{i,\ \deg(v_i) + 1\}) \tag{3.4}$$

is the upper bound on the number of colors used by the Sequential algorithm to color the vertices $v_1, v_2, \ldots, v_n$ of graph G. This holds true irrespective of the ordering of the vertices of the graph G.

There are various orderings of the vertices of G which have been considered for applying the Sequential algorithm. An obvious beginning was made with the LF ordering in which vertices with the largest degrees were ranked first in the order *i.e.* $\deg(v_1) \geq \deg(v_2) \geq \ldots \geq \deg(v_n)$. The sequential algorithm run on such an ordering is called the *largest first sequential* algorithm (LFS algorithm). An LF ordering is superior to the other orderings when they are used in the sequential algorithm. This fact is stated as Theorem (3.2) below, again quoted from Syslo *et al* [24].

[Theorem 3.2] Let $u_1, u_2, \ldots, u_n$ be an LF ordering of the vertices of a graph G and $u_S(G;\ v_1,\ v_2,\ \ldots,\ v_n)$ be the upper bound on the number of colors used by the Sequential algorithm for coloring G with any arbitrary ordering of it's vertices. Let $u_{LF} = u_S(G;\ u_1,\ u_2,\ \ldots,\ u_n)$ be the upper bound on the number of

colors used by the Sequential algorithm for coloring G with an LF ordering $u_1, u_2, \ldots, u_n$ of it's vertices. Then

$$u_{LF} = \text{Min } (G; \Pi_1, \Pi_2, \ldots, \Pi_n)$$

where the minimum is taken over all orderings $\Pi_1, \Pi_2, \ldots, \Pi_n$ of the vertices of G. ■

(The proof, which is omitted here, is outlined in Syslo *et al* [24].)

Another ordering which is similar to LF ordering is the SL (*smallest-last*) ordering. Such an ordering is constructed for the vertices of graph G using the following procedure described by Matula [26] :-

(i) v_n is a minimum degree vertex of G.

(ii) For $i = n-1, n-2, \ldots, 2, 1$,

v_i is a minimum degree vertex in the subgraph of G induced by $V - \{v_n, v_{n-1}, \ldots, v_{i+1}\}$.

The ordering is called smallest-last because

$$\deg_i(v_i) = \min_{1 \le j \le i} \{\deg_i(v_j)\} \tag{3.5}$$

where $\deg_i(v_i)$ is the degree of v_i in the subgraph induced by $v_1, v_2, \ldots, v_i$. The sequential coloring algorithm applied to a SL ordering is called the SLS algorithm.

A careful examination of the Sequential Coloring algorithm reveals that a vertex v_i can be colored by one of the first i colors *or* by one of the first $\deg_i(v_i) + 1$ colors, whichever is minimum. Hence the upper bound $u_S(G;\ v_1,\ v_2,\ \ldots,\ v_n)$ on the number of colors used by the Sequential algorithm can be improved to

$$u_S'(G;\ v_1,\ v_2,\ \ldots,\ v_n) = \max_{1 \le i \le n} \ \min\ \{i,\ \deg_i(v_i) + 1\} \tag{3.6}$$

Since $\deg_i(v_i) \le \deg(v_i)$ for any ordering $v_1,\ v_2,\ \ldots,\ v_n$ of the vertices, therefore

$$u_S'(G;\ v_1,\ v_2,\ \ldots,\ v_n) \le u_S(G;\ v_1,\ v_2,\ \ldots, v_n). \tag{3.7}$$

Let $u_{SL} = u_S'(G;\ t_1,\ t_2,\ \ldots,\ t_n)$ be the upper bound on the number of colors used by the Sequential algorithm for coloring G with an SL-ordering of it's vertices $t_1,\ t_2,\ \ldots,\ t_n$. By the nature of construction of SL-ordering of the vertices,

$$u_{SL} = \min\ u_S'(G;\ \Pi_1,\ \Pi_2,\ \ldots,\ \Pi_n), \tag{3.8}$$

where the minimum is taken over all the orderings $\Pi_1,\ \Pi_2,\ \ldots,\ \Pi_n$ of the vertices of G.

By (3.7), $\quad u_S'(G;\ u_1,\ u_2,\ \ldots,\ u_n) \le u_S(G;\ u_1,\ u_2,\ \ldots,\ u_n),$

$$\tag{3.9}$$

where $u_1,\ u_2,\ \ldots,\ u_n$ is a LF-ordering of the vertices of G.

By (3.8), $\quad u_S'(G;\ t_1,\ t_2,\ \ldots,\ t_n) \le u_S'(G;\ u_1,\ u_2,\ \ldots,\ u_n)$

(3.10)

Therefore,

$$u_{SL} = u_S'(G;\ t_1,\ t_2,\ \ldots,\ t_n) \le u_S(G;\ u_1,\ u_2,\ \ldots,\ u_n) = u_{LF},$$

i.e., $$u_{SL} \le u_{LF} \tag{3.11}$$

which suggests that the SLS algorithm is superior to the LFS algorithm.

There are several other heuristic algorithms for graph coloring which are sequential in nature. An important one amongst these is the SLF (*saturation-largest-first*) algorithm, also known as the DSATUR algorithm, due to Brélaz [27].

[Definition] A *saturation degree* of an uncolored vertex v in a partially colored graph G is defined as the number of different colors to which v is adjacent. ■

The SLF algorithm proceeds as follows :-

(i) Color a vertex of minimum degree in graph G with color 1.

(ii) For i = 2, 3, ..., n do
As v_i take an uncolored vertex of maximum saturation degree breaking ties, if any, by choosing the vertex of greater degree and color v_i with the least possible

color.

3.1.2.2 Sequential Algorithms with Interchange (SI)

The simple sequential algorithm can be improved by re-arranging, at each step in the coloring sequence, the colors of the already colored vertices to avoid, if possible, using an additional color for the next vertex to be colored. The interchange procedure in Sequential algorithms was first described by Matula *et al* [26].

Assume that $v_1, v_2, \ldots, v_{i-1}$ have been colored by k colors and the simple sequential algorithm colors v_i with color $k+1$. This implies that v_i has neighbours amongst the vertices $v_1, v_2, \ldots, v_{i-1}$ colored with $1, 2, \ldots, k$. If the subgraph formed by these neighbours of v_i contains a complete subgraph on k vertices then the new color $k+1$ is necessary for v_i. Otherwise, it may be possible to interchange the colors of some neighbours of v_i, preserve the k-coloring of the subgraph generated by $v_1, v_2, \ldots, v_{i-1}$ and free one of the first k colors for v_i.

Let G_{pq} be the subgraph induced by the vertices colored with color p or q, $1 \le p, q \le k$. If G_{pq} has two vertices u and v of different colors, both adjacent to v_i and both belonging to the same connected component of G_{pq}, then no interchange of p and q can free one of these two colors for v_i. Therefore, if this

holds for every p and q, v_i must be colored with k+1. On the other hand, if there exist p and q such that in every connected component of G_{pq}, the vertices adjacent to v_i are of at most one color, then the (p,q)-interchange procedure, described below, can be applied fruitfully.

In the (p,q)-interchange procedure let U_p be the set of vertices of those connected components of G_{pq} which contain a neighbour v of vertex v_i, such that *COLOR*(v) = p. Now, for every $u \in U_p$, if *COLOR*(u) = p then set *COLOR*(u) = q, and if *COLOR*(u) = q then set *COLOR*(u) = p.

It is easily seen that while the coloring of v_1, v_2, ..., v_{i-1} remains feasible, color p has become free for v_i. The interchange procedure, as described above, is an extension of the original version due to Matula *et al* [26] and it initially appeared in a paper by Johnson [28].

Obviously the SI algorithms can be applied to any ordering of the vertices of a graph, for example one can have LFSI and SLSI algorithms.

Despite the claim that the SI algorithm is an improvement over the Simple Sequential algorithm, it is possible that the SI algorithm uses more colors than the Simple Sequential algorithm for some random graphs and orderings. This is particularly true if the interchange of colors is carried out

early in the sequential coloring procedure. In general, all the approximate coloring algorithms following the sequential approach are capable of producing very bad colorings, in terms of number of colors used, for certain graphs. On the average, however, they seem to fare reasonably well, both in terms of running time as well as number of colors used.

3.1.2.3 Sequential Algorithm with Backtracking (SB)

The idea of sequential coloring can be extended to give rise to a backtracking method which will color every graph optimally *i.e.* with the minimum number of colors. The backtracking sequential algorithm was first proposed by Brown [29].

Consider a graph G with an arbitrary ordering of it's vertices $v_1, v_2, \ldots, v_n$. Let $v_1, v_2, \ldots, v_{i-1}$ be vertices which are already colored. For the vertex v_i, find a set U_i of feasible colors and assign Min $\{U_i\}$ to vertex v_i. Let l_{i-1} denote the maximum color number used to color the vertices $v_1, v_2, \ldots, v_{i-1}$. Every color j that belongs to U_i must satisfy each of the following conditions :-

(i) $j \leq l_{i-1} + 1$, so that no redundant colorings are generated at any stage.

(ii) $j \leq \text{Min}\ \{i, \deg(v_i) + 1\}$, as per (3.1)

(iii) j is not a color of any vertex v_h $(1 \le h \le i-1)$ which is a neighbour of v_i.

(iv) If a complete q-coloring of G is known, then $j \le q - 1$.

The coloring procedure backtracks from vertex v_i to vertex v_{i-1} whenever $U_i = \phi$ at any stage. This is done by deleting from U_{i-1} the current color of v_{i-1} and restarting the coloring procedure from the vertex v_{i-1} using the new U_{i-1}. The procedure exits if a backtrack is required from vertex v_1 implying that no coloring better than the already known q-coloring can be found. The procedure also exits when v_n is the last vertex colored and this implies that a l-coloring has been found, where $l < q$. Set $q = l$ and try to get a fresh coloring by repeating the backtracking procedure afresh.

The backtracking algorithm described above may be refined by using an intelligent ordering of the vertices. If instead of an arbitrary ordering, the first h vertices are so ordered that they form a clique (each of the vertices in a clique is adjacent to all the others), which is the largest in G, followed by the remaining vertices v_i in such an ordering such that v_i is adjacent to more of the vertices v_1, v_2, ..., v_{i-1} than any other vertex v_j for $j > i$, then the backtracking need not go back farther than v_{h+1}. In other words, it is not necessary to backtrack to v_1 for enumerating all the

non-redundant colorings. So the algorithm becomes more efficient. However, finding the largest clique in a graph is also a NP-Complete problem like the Graph Coloring Problem; hence a greedy method of ordering the vertices due to Brown [29] is chosen. In this ordering, called the *Greedy Largest First Ordering* (GLF), the first vertex v_1 is chosen to be of the largest degree in G. Subsequent vertices v_i are such that v_i is adjacent to more of the vertices v_1, v_2, ..., v_{i-1} than any other vertex v_j for $j > i$.

Look-ahead procedures can also be incorporated into the backtracking algorithm to improve it's efficiency. These look a few steps ahead and check the effect of a current color assignment on a later stage and if any infeasibility is detected early on, the current color assignment can be dropped and the next color can be considered for assignment right away. Before applying such methods, the additional costs of computation for a look-ahead procedure must be balanced against the savings in enumeration it generates for the backtracking algorithm. The look-ahead procedures for sequential backtracking algorithm have been explored by Brown [29] and Korman [30].

3.2 Partitioning the graph into easily colorable graphs

The technique of solving a problem by breaking it into smaller, more easily solvable parts, solving these partial

problems, and then combining the partial solutions into a solution to the whole problem is called *Divide-and-Conquer*. Applied to the design of algorithms, the method involves partitioning a problem into smaller, homogenous pieces, to which the overall technique is then applied recursively. If the parts can be combined efficiently, the method may lead to an efficient algorithm. *Partitioning* is a related technique, in which a problem is divided into pieces, solutions are obtained for each piece, and are then combined to form a solution for the whole problem. The rather thin difference between divide-and-conquer and partitioning is that divide-and-conquer typically involves recursion, while partitioning does not. However, this difference will be ignored in the following text.

The divide-and-conquer technique is used in many graph algorithms. To apply this technique, the input graph is decomposed into a hierarchy of components. Then the problem is solved on each of the components at the bottom of the hierarchy and gradually the solutions on larger and larger components are pieced together, until a solution is finally found for the entire graph.

Usually the decomposition that supports divide-and-conquer is based on finding *separators* of the smallest size. Separators are those entities of the graph whose elimination results in the decomposition of the graph. Different kinds of separators can be considered for decomposing a graph :-

(i) Vertex set - elimination of a set of vertices from the graph partitions it in two or more components. (Figure 3.1)

(ii) Edge set - elimination of a set of edges results in a disconnection of the graph into two or more components. This is equivalent to generating a s-t cut in the graph, where s and t are vertices in the two different components. (Figure 3.2)

(iii) Clique - elimination of a clique results in the decomposition of the graph into two or more components. (Figure 3.3)

The separator chosen for the partition of the input graph into easily colorable graphs is the Edge set. The basis for this choice lies in the fact that the interference phenomena is of a local nature and the edges, in the graph GC to be colored, are representative of the constraints on coloring (frequency assignment) due to the interference effect. So by partitioning the graph GC while choosing Edge sets of least cardinality, the induced subgraphs are inter-connected by minimal number of edges and hence have the maximum de-coupling possible. In such a situation, the effect of coloring (frequency assignment) within an induced subgraph will have minimal effect over the coloring (frequency plan) for the other induced

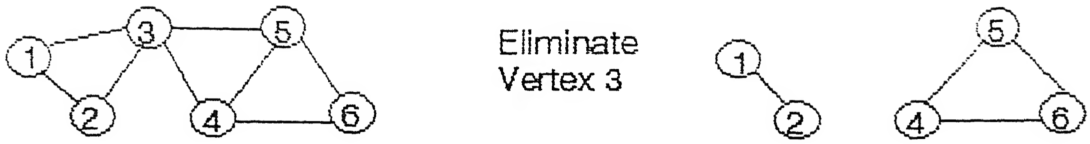


FIGURE 3.1 Vertex separator

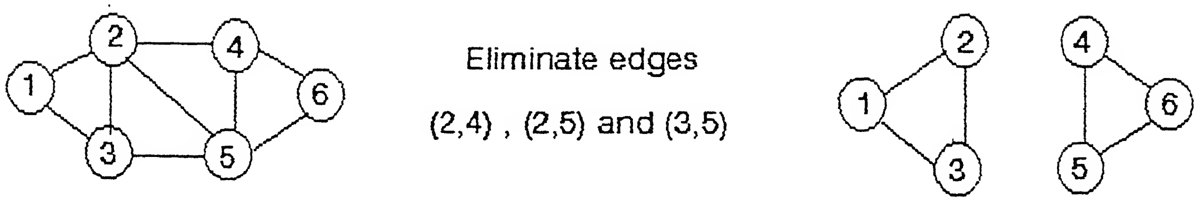


FIGURE 3.2 Edge separators

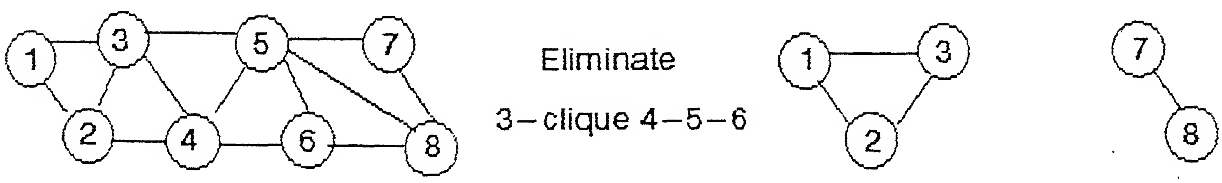


FIGURE 3.3 Clique separator

subgraphs. As a result the amount of computational work required to combine the individual colorings of_all the induced subgraphs and get the coloring for the entire input graph, is kept as low as possible.

The procedure for generating the Edge set is based on the Max Flow - Min Cut Theorem due to Ford and Fulkerson [31]. All the edges in the input graph GC are assumed to have unit capacity and so the capacity of the minimum cut is equal to the number of edges present in the minimum cut. The edges in the minimum cut comprise the Edge set.

[Definition] *Edge connectivity* between two subgraphs G_1 and G_2, induced by two disjoint subsets V_1 and V_2 of the vertices of a graph G, is the total number of edges in G joining a vertex in G_1 to a vertex in G_2. ∎

The above definition implies that, given two subgraphs G_1 and G_2 induced by two disjoint subsets of vertices such that the edge connectivity between them is K, the number of edges that have to be cut in order to disconnect any two vertices - one in G_1 and the other in G_2 - is K or less. Given a specified maximum value of the edge connectivity K between any two subgraphs induced by the partition, it is desired that the maximum flow (or minimum cut) value between any two vertices in the same induced subgraph exceeds the value K *i.e.* to disconnect any two vertices in the same induced subgraph, more than K edges have to be cut.

The partitioning procedure relies upon the knowledge of maximum flow values for all pairs of vertices in the original graph, which in turn can be computed efficiently by constructing a Gomory-Hu Cut-Tree [32].

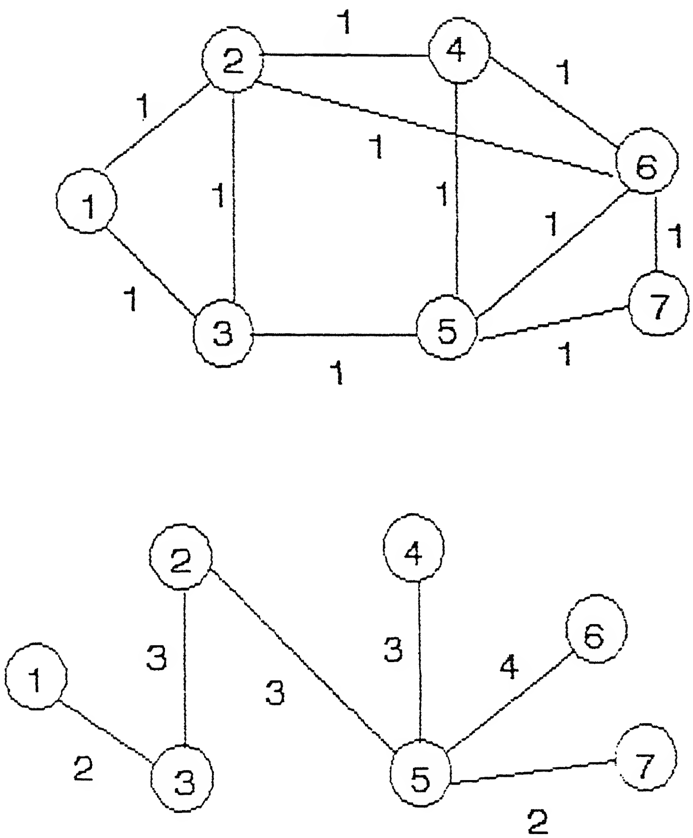
3.2.1 Multi-terminal maximum flows and Graph Partitioning

The Gomory-Hu cut-tree which is used to compute multi-terminal maximum flows *i.e.* maximum flow values for more than one pair of vertices, has the following properties :-

(i) It is flow-equivalent to the original graph from which it is derived. Two networks (graphs or trees) are called flow-equivalent if there is one-to-one relation between the vertices of the two networks and the maximum flow values between any two of the vertices is the same in both the networks.

(ii) Each link of the tree represents a minimum cut of the original graph. This is also the reason why it was called a cut-tree by Gomory and Hu.

(iii) The complete Gomory-Hu cut-tree can be constructed by solving O(n) maximum flow problems where n is the number of vertices in the graph. (The process of construction of a complete Gomory-Hu cut-tree is described under Section 2.2)



A given graph GC

Equivalent tree

network which is a

Gomory–Hu cut–tree

FIGURE 3.4 Flow equivalence of a given graph to it's Gomory–Hu cut–tree

(Figure 3.4 illustrates the Gomory-Hu cut-tree derived from a given graph and the flow equivalence of the two.)

The maximum flow value between any two vertices is equal to the capacity of the unique path between the two vertices in an equivalent tree network. Using a predecessor index representation of the equivalent tree network, the path between any two vertices can be determined quite efficiently and hence, the maximum flow between the two vertices can also be computed easily. Therefore, the maximum flow values for all pairs of vertices (there will be $O(n^2)$ pairs) can be found out by solving $O(n)$ maximum flow problems, instead of $O(n^2)$, by virtue of property (iii) stated above. Given the maximum flow values for all pairs of vertices, the graph may be partitioned along following lines :-

ALGORITHM PARTITION

STEP 1 : $I := 1$

CONSIDER a vertex v from V /* V is the vertex set of the graph to be partitioned */

$P_I := \{v\}$ /* P_I is the I^{th} partition subset of V */

$V := V - \{v\}$

READ(K) /* K is the specified edge connectivity between any two partitions */

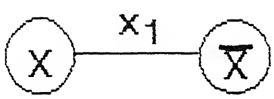
STEP 2 : *IF* $V \neq \phi$ *THEN*

```
                    CONSIDER a vertex u from V
                    J := 1
STEP 2.1   :        IF J ≤ I THEN
                         IF maximum-flow(u,p_J) > K for every
                              vertex p_J ∈ P_J THEN P_J := P_J ∪ {u}
                         ELSE J := J + 1
                              GOTO / 2.1
                    ELSE I := I + 1
                         P_I := {u}
                         GOTO / 2
STEP 3     :    FOR J := 1  TO  I  OUTPUT(P_J)     /* P_J is the J^th
                    partition subset of V, the vertex set of  the
                    graph */
```

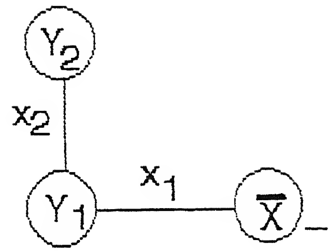
3.2.2 Construction of a Gomory-Hu cut-tree

The partitioning algorithm described in Section 2.1 utilized the maximum flow values between all pairs of vertices, which in turn could be calculated efficiently by using the complete Gomory-Hu cut-tree. The process of construction of a complete Gomory-Hu cut-tree (a few stages are shown in Figure 3.5) involves the following steps :-

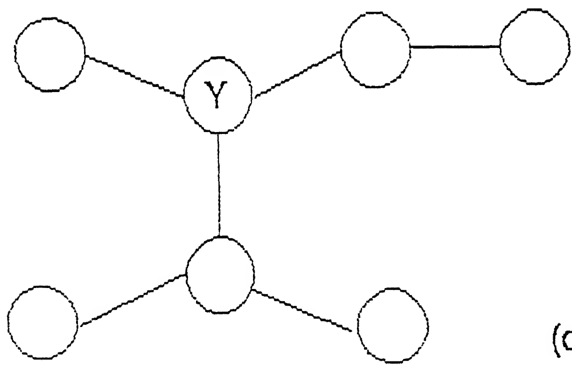
STEP (i) Select two vertices - SOURCE and SINK - arbitrarily in the original graph and compute the maximum flow between them using the original graph.



) First partition of the graph into bignodes X and $\overline{X}$. Maximum flow between SOURCE and SINK is x_1.



(b) Second partition of the graph by splitting bignode X into Y_1 and Y_2.



(c) Bignode Y in an intermediate stage

FIGURE 3.5 Stages in the construction of a Gomory–Hu cut–tree

STEP (ii) Use the resultant minimum cut to partition the vertices into two parts, SOURCE and SINK being in different partitions. Label each partition as a Bignode along with a unique index number attached to identify any bignode.

STEP (iii) Join the two bignodes obtained above by an edge having a capacity equal to the maximum flow value computed in (i). This edge is the first link of the cut-tree. The cut-tree also has two bignodes now.

STEP (iv) Select any bignode in the cut-tree developed so far, which contains two or more vertices. Repeat steps *(i)-(ii)-(iii)* by choosing a SOURCE and a SINK from the bignode selected.

If no bignode can be selected in step *(iv)* *i.e.* all bignodes contain only one vertex then the process halts. The cut-tree developed at the end of the process is the required Gomory-Hu cut-tree.

For computing the maximum flow between SOURCE and SINK use a simplified network derived in the following manner :- Suppose the bignode selected in step *(iv)* is deleted from the cut-tree developed so far. This results in a disconnection of the cut-tree into various components. All the vertices belonging to any of the bignodes in one component are condensed into a single node. The vertices belonging to the selected bignode are

not condensed and the edges between them are copied from the original graph. Edges are drawn between two condensed nodes or a condensed node and an original vertex. The capacity of an edge drawn between two condensed nodes is equal to the sum of the capacities of the edges in the original graph joining a vertex in one condensed node to a vertex in the other condensed node. Similarly, the capacity of an edge drawn between a condensed node and an original vertex is equal to the sum of the capacities of the edges in the original graph joining a vertex in the condensed node to the original vertex. (The rationale behind the process of condensing a subset of vertices into a single vertex for the purposes of simplifying the maximum flow computations is explained in Section 2.3)

It is seen that the above procedure for constructing a cut-tree creates a partition of the vertices in each iteration. By allowing a partition in step *(ii)* only when the maximum flow value computed in step *(i)* is smaller than K, the cut-tree obtained at the end of the procedure will contain bignodes which are partitions such that the maximum flow between any two vertices contained in a bignode is greater than K. This being the case, there is no need to go for a separate partitioning algorithm like the one described under Section 2.1. However, it should be kept in mind that such a short-cut procedure may require, for each bignode to be partitioned, maximum flow computations for more than one pair of vertices in the bignode. If there are n_i vertices in the i^{th} bignode, $O(n_i^2)$ maximum flow

computations may be needed for resolving the problem of partitioning the i^{th} bignode. On the other hand, the complete cut-tree can be constructed by solving n-1 maximum flow problems. The short-cut procedure does, however, provide an alternative to the earlier scheme of partitioning, based on processing the complete cut-tree to get maximum flow values for all pairs of vertices, and it's implementation may be more efficient in some cases. The cut-tree generated at the end of the procedure with the above modification is called a partial cut-tree of edge connectivity K. The algorithm for partial cut-tree can also be used to generate the complete cut-tree by setting the parameter K = n-1. The algorithm for constructing a partial Gomory-Hu cut-tree of specified edge connectivity K, is derived by modifying the procedure given by Gomory and Hu [32] for developing a complete cut-tree. The algorithm for constructing the partial cut-tree is given below.

ALGORITHM PARTIAL_CUT_TREE

DATA STRUCTURES

C_TREE : Vertex adjacency list format for storing the cut-tree

WGR : Vertex adjacency list format for storing the working graph. The working graph is derived from the input graph GC with some subsets of the vertices condensed to single vertices, wherever possible, for the purpose of Maximum-Flow computations

MATRIX_MAXFLOW : A n X n matrix to store Maximum-Flow values between all pairs of vertices (SOURCE-SINK pairs)

BIGNODE : A node of C_TREE which contains all the vertices of the partition induced subgraph it denotes

N_VERT : Number of vertices in WGR

CONTINUE_SPLITTING : A boolean variable to store a decision

STEP 1 *INITIALIZE* C_TREE

STEP 2 *INITIALIZE* WGR

STEP 3 *IF* N_VERT > 1 *THEN* CONTINUE_SPLITTING := true

STEP 4 *WHILE* CONTINUE_SPLITTING *DO*

STEP 4.1 *SELECT* SOURCE and SINK vertices

IDENTIFY the BIGNODE in C_TREE which contains SOURCE and SINK vertices

IF unable to make any selection *THEN*

CONTINUE_SPLITTING := false

STEP 4.2 *IF* CONTINUE_SPLITTING *THEN*

STEP 4.2.1 *CONDENSE* all disjoint subsets of vertices not in BIGNODE into single vertices in WGR, wherever possible

STEP 4.2.2 *DO* MAXIMUM-FLOW computation on WGR between SOURCE and SINK to get MAXFLOW value

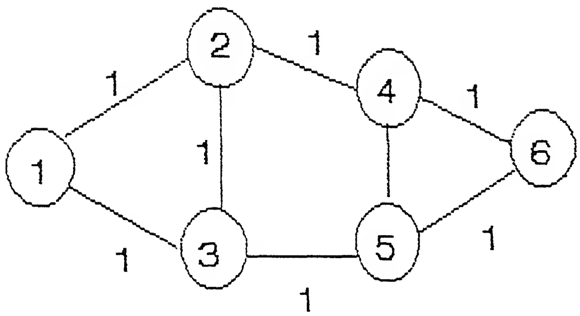
STEP 4.2.3 *UPDATE* values in MATRIX_MAXFLOW for the SOURCE-SINK pair

STEP 4.2.4	*IF* MAXFLOW < K *THEN MODIFY* C_TREE *i.e.* create a partition induced subgraph in the form of additional bignodes in C_TREE
STEP 4.2.5	*RE-INITIALIZE* WGR

3.2.3 Condensing a subset of vertices into a single vertex

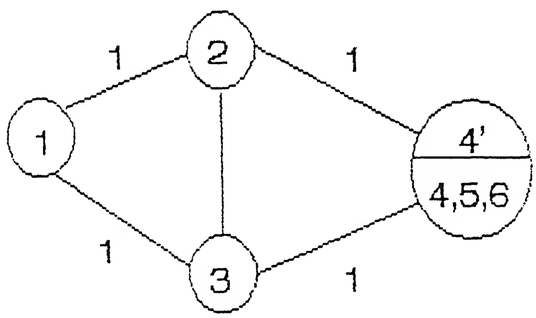
Maximum flow computations required to be done in the *ALGORITHM* PARTIAL_CUT_TREE, typically consume a time bounded by a polynomial function of the number of vertices n and the number of edges m in the graph. For example, Karzanov's maximum flow algorithm runs in $O(n^3)$ time. The maximum flow computations can be done on a simplified graph with lesser number of vertices, called the *working graph*. This will result in a reduction in the number of computations to be made. The working graph WGR is derived from the original graph GC by utilising a process of condensing a subset of vertices into a single vertex. An example is illustrated in Figure 3.6.

By condensing 4, 5 and 6 into a single vertex the edge capacities of (4,5), (4,6) and (5,6) have been raised from 1 to infinity. Since these edges are not in the 2-3 minimum cut, any increase in their capacities does not affect the maximum flow value between the vertices 2 and 3. Therefore, so far as calculating the maximum flow value between the vertices 2 and 3 is concerned, the vertices 4, 5 and 6 may be condensed into a



(a) Original graph GC

Condense vertices 4.5 and 6



(b) Simplified graph WGR

NOTE :– The maximum flow between 2 and 3 is 3 units and it is the same in both GC and WGR.

FIGURE 3.6 Condensing a subset of vertices into a single vertex.

single vertex and the computations may be done on the graph WGR.

There are a couple of conditions under which the process of condensation is allowed and they are stated below in the form of Lemma (3.1) and Lemma (3.2) given in [Hu,*].

[Lemma 3.1] Let $(X, \overline{X})$ be a minimum cut separating $v_i \in X$ and some other vertex, and let v_e and v_k be any two vertices contained in $\overline{X}$. Then there exists a minimum cut $(Z, \overline{Z})$ separating v_e and v_k such that $(Z, \overline{Z})$ and $(X, \overline{X})$ do not cross each other. ■

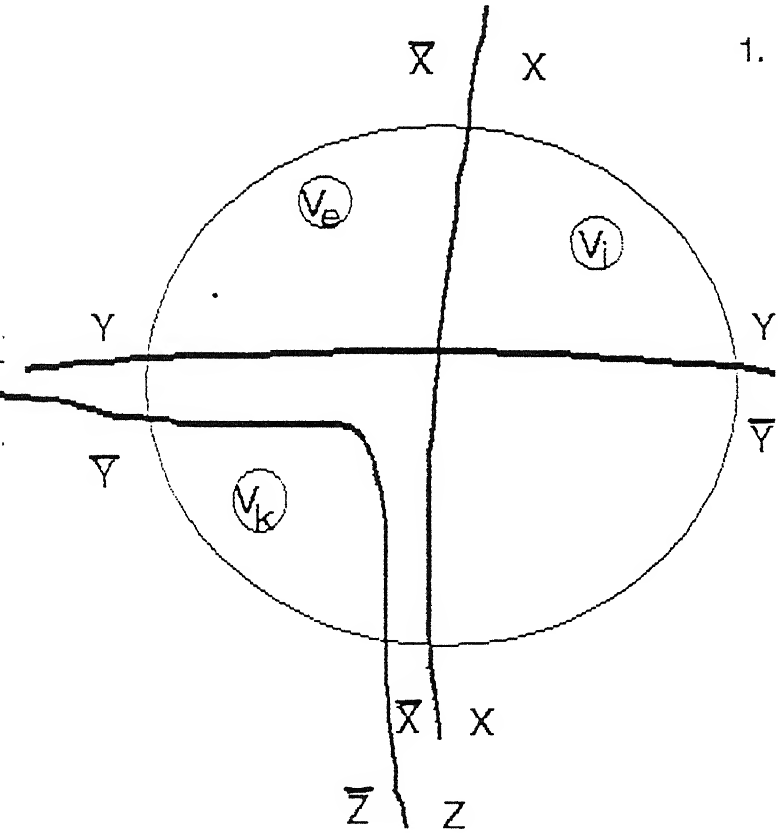
(Refer to Figure 3.7 for details of the cuts.)

This means that if $(X, \overline{X})$ is a minimum cut, the subset X can be condensed into a single vertex for computing the maximum flow between any two vertices in $\overline{X}$.

[Lemma 3.2] Let $(X, \overline{X})$ be a minimum cut separating v_i and some other vertex, and let v_k be any vertex that belongs to $\overline{X}$. Then there exists a minimum cut $(Z, \overline{Z})$ separating v_i and v_k such that $(X, \overline{X})$ and $(Z, \overline{Z})$ do not cross each other. ■

(Refer to Figure 3.7 for details of the cuts.)

This means that if $(X, \overline{X})$ is a minimum cut separating v_i and some other vertex and the value of the maximum flow



1. $(Y,\overline{Y})$ and $(X,\overline{X})$ are minimum cuts which cross each other.

2. $(Z,\overline{Z})$ and $(X,\overline{X})$ are non–crossing minimum cuts.

FIGURE 3.7 Illustration of Lemma 3.1 and Lemma 3.2.

between i and k is to be determined, where v_k is any vertex in $\overline{X}$, then the subset X can be condensed into a single vertex. The *STEP* 4.2.1 in *ALGORITHM* PARTIAL_CUT_TREE for condensation is carried out based on the implications of Lemma (3.1) and Lemma (3.2) to get a simplified working graph WGR. The maximum flow computations are carried out on WGR to effect some savings in the amount of computation required.

3.2.4 Algorithm for Maximum Flow computation

Any of the standard routines, surveyed below, can used for computing maximum flow between a source vertex and a sink vertex. The basic steps of a maximum flow algorithm are :-

(i) Based on the current flow, the residual network is constructed. Each arc in the residual network has a non-zero capacity, called residual capacity = capacity of the arc in the original network - flow along the arc.

(ii) A flow augmenting path (path along which flow can be sent) between the source vertex and the sink vertex in the residual network is selected and flow is sent along this path.

For arbitrary selection of the flow augmenting path,

the running time of the above algorithm can go upto $O(nmU)$ where n is the number of vertices, m is the number of edges in the graph and U is the upper bound on the arc-flow capacities. The first major improvement in algorithmic efficiency came from Edmonds and Karp who selected the shortest flow augmenting path (*i.e.* the path with the least number of arcs) every time in step (ii). The running time of this algorithm is bounded by $O(nm^2)$. There are many algorithms which improve upon this bound. Most of these are based on the algorithm given by Dinic which uses the concept of layered networks. Dinic's algorithm runs in $O(n^2m)$ time. A major improvement upon this was given by Malhotra *et al*. Their MPM algorithm has a running time bounded by $O(n^3)$. A fundamentally different approach using the method of pre-flows is used by Karzanov's algorithm which is also $O(n^3)$ in computational complexity. The technique of capacity scaling adopted by Tarjan *et al* has further pushed the bound which is practically achievable to $O(nm \log(n^2/m))$.

Keeping in view that most of the time arc-flow capacities in the graph GC are small numbers, the basic flow augmenting path algorithm can be expected to do well from a computational point of view. The implementation of this algorithm is also considerably easier.

3.3 Merging Algorithm for Colored Subgraphs

Any generalized algorithm for merging two colored subgraphs can be used to merge more than two colored subgraphs by an iterative process. The generalized algorithm for merging two colored subgraphs GR_A and GR_B takes them as input along with a list L of edges which join a vertex in GR_A to a vertex in GR_B in the original graph GC. The output is a colored subgraph GR_A_B induced by the vertices of GR_A and GR_B. In general, GR_A_B uses more colors than GR_A and GR_B. While the ordering of the edges in the input list L may be arbitrary, the number of colors added during the merging process depends upon the process itself and the ordering of the vertices in the list L.

The algorithm is structured such that each edge in the list L has a head which is a vertex of GR_A and a tail which is a vertex of GR_B. Each edge in the list L is examined once in the order in which it appears. Initially, the coloring for each vertex in GR_A and GR_B is unconstrained *i.e.* it is free to be changed during the process of merging. One of the following steps is taken upon examining each edge in the list L :-

(i) If *COLOR*(head) ≠ *COLOR*(tail) then the coloring for the head and the tail are constrained to their respective colors.

(ii) If *COLOR*(head) = *COLOR*(tail) and coloring of both head

and tail are constrained to their respective colors then an additional color is used for the head (or alternatively the tail).

(iii) If *COLOR*(head) = *COLOR*(tail) and coloring of either head or tail is constrained, say that of head, but not both or coloring of neither is constrained then the unconstrained colors in the subgraph GR_B (which contains the tail) can be re-arranged to get a re-coloring of the tail which does not conflict with the coloring of the head.

```
ALGORITHM MERGING
STEP 1    :   (Initialization)
              FOR I := 1 TO N DO CONS_A[I] := FALSE
                                 CONS_B[I] := FALSE
              /* Color of vertex I in GR_A and GR_B is
                 unconstrained initially */
              TEMPEDGE := POINTER_TO_TOP_OF_LIST(L)
              /* L is the list of edges which join a a vertex in
                 GR_A to a vertex in GR_B in the original graph
                 GC, HEAD being in GR_A and TAIL in GR_B */
              COLOURS_A := NUMBER_OF_COLORS_IN_GR_A
STEP 2    :   (Body of the algorithm)
              IF TEMPEDGE ≠ NIL   /* List L is not empty */
              /* One of the following four cases occurs */
              CASE 1 ( COLOR(HEAD(TEMPEDGE)_IN_GR_A) ≠
```

```
                    COLOR(HEAD(TEMPEDGE)_IN_GR_A) )
          CONS_A[HEAD(TEMPEDGE)] := TRUE
          CONS_B[TAIL(TEMPEDGE)] := TRUE
     /* Both head and tail  are  constrained  to  their
        respective colors */
     CASE 2 ( COLOR(HEAD(TEMPEDGE)_IN_GR_A) =
                    COLOR(HEAD(TEMPEDGE)_IN_GR_A) ) AND
            CONS_A[HEAD(TEMPEDGE)] = TRUE AND
            CONS_B[TAIL(TEMPEDGE)] = TRUE
          COLOR(HEAD(TEMPEDGE)_IN_GR_A) := COLOURS_A+1
     /* An additional color is used */
     CASE 3 ( COLOR(HEAD(TEMPEDGE)_IN_GR_A) =
                    COLOR(HEAD(TEMPEDGE)_IN_GR_A) ) AND
            CONS_A[HEAD(TEMPEDGE)] := TRUE AND
            CONS_B[TAIL(TEMPEDGE)] = FALSE
          CALL RENAME_COLOR(CONS_B,GR_B)
     CASE 4 ( COLOR(HEAD(TEMPEDGE)_IN_GR_A) =
                    COLOR(HEAD(TEMPEDGE)_IN_GR_A) ) AND
            CONS_A[TAIL(TEMPEDGE)] = FALSE
          CALL RENAME_COLOR(CONS_A,GR_A)
     TEMPEDGE := POINTER_TO_NEXT_OF_TEMPEDGE
     GOTO / 2
     /* Next edge in the list L is examined */


PROCEDURE RENAME_COLOR(CONS,GR)
STEP 1    :    (Initialization)
     FOR I := 1 TO N DO C[I] := 0
```

FOR I := 1 *TO* N *DO* C[COLOR(VERETX_I_IN_GR)] :=

COLOR(VERETX_I_IN_GR)

FOR I := 1 *TO* N *DO* *IF* CONS[I] *THEN*

C[COLOR(VERETX_I_IN_GR)] := 0

STEP 2 : (Rearrange colors if possible else use additional color)

IF C[I] ≠ 0 for more than one value of I *THEN*

Rearrange in a cyclical order the values of C[I] amongst those I for which C[I] ≠ 0

COLOR(VERTEX_IN_GR) := C[I]

/* Rearrange colors of vertices of GR */

ELSE Use an additional color for HEAD in GR_A as in *CASE* 2 of the main body of the algorithm.

3.4 Evaluation of Intermodulation Interference

As explained in Chapter II, evaluation of the order of intermodulation interference is to be carried out at each site in the communication network for a given frequency plan. The evaluation problem was formulated as a nonlinear integer program (2.9) :-

$$\text{Min} \quad Q_o = \sum_{i=1}^{n} |X_i|$$

$$\text{Subject to} \quad \sum_{i=1}^{n} F_i X_i \leq GB;$$

$$\sum_{i=1}^{n} F_i X_i \geq -GB;$$

$\exists$ at least one i such that $|X_i| = 1$;

X_i are integers and free;

where $1 \le i \le n$.

To declare the set of frequencies F_1, F_2, ..., F_n, which are active at a site in the network, to be intermodulation-free upto order Q (or less), the optimal value of Q_o *i.e.* Q_o^{opt} obtained by solving (.) should be greater than Q+1. As there is no real purpose in determining Q_o^{opt} other than finding out whether $Q_o^{opt} >$ Q+1 or not, a bound on the value of Q_o may be placed through an additional constraint

$$Q_o \le Q + 1 \tag{3.12}$$

If there is no feasible solution to (2.9) along with the above constraint then it means that $Q_o^{opt} > Q+1$, else $Q_o^{opt} \le Q+1$.

The above nonlinear integer program can be linearised easily by introducing fresh variables X_i^+ and X_i^- such that

$$X_i = X_i^+ - X_i^- \tag{3.13}$$

The resulting linear integer program can be written as

$$\text{Min} \sum_{i=1}^{n} (X_i^+ + X_i^-) \tag{3.14}$$

$$\text{Subject to} \quad \sum_{i=1}^{n} F_i (X_i^+ - X_i^-) \le GB; \tag{3.15}$$

$$\sum_{i=1}^{n} F_i (X_i^+ - X_i^-) \ge -GB; \tag{3.16}$$

$$\exists \text{ at least one } i \text{ such that } X_i^+ + X_i^- = 1; \tag{3.17}$$

$$X_i^+ , X_i^- \geq 0 \text{ and integers, where } 1 \leq i \leq n.$$

The optimal solution of the LP-relaxation of the above problem is

one of the $X_i^+ , X_i^- = 1$;

$$X_j^+ = \frac{F_i}{F_j} \text{ ; and} \qquad (3.18)$$

rest of the $X_k^+ , X_k^- = 0$, where $k \neq i, j$ and,

F_j is the highest frequency among the n-1 frequencies excluding F_i; and

$$Q_o^{LP\text{-}opt} = 1 + \frac{F_i}{F_j} \qquad (3.19)$$

In most practical cases of intermodulation problems, $Q_o^{LP\text{-}opt}$ is substantially smaller than the optimal integer solution Q_o^{opt}. Therefore any attempt at a solution by using a branch and bound procedure runs into the difficulty of obtaining a good lower bound based on the LP-relaxation. So backtrack algorithms are used as a solution strategy.

In the primal backtrack algorithm, an initial solution is found and then improved solutions are sought. The procedure terminates when no solution with a lower Q_o-value can be found.

The dual backtrack algorithm which has been used to solve the problem (2.9) along with the constraint (3.12), starts with $Q_o = 1$ and tries to find a feasible solution to the problem for the fixed value of Q_o, failing which it increments Q_o by 1

and repeats the solution procedure until it finds a solution or $Q_o = Q + 1$. If it has found no solution upto the stage $Q_o = Q+1$ then it exits with the inference that $Q_o^{opt} > Q+1$. The first solution, if found, is also the optimal since it is the dual problem which is being solved. The algorithm runs as follows :-

ALGORITHM DUAL_BACKTRACK

STEP 1 : (Initialization)

GOAL := 1;

X_1 := - GOAL;

$X_2, X_3, \ldots, X_n$:= 0;

LEVEL := 2;

ONE := 1; { ONE indicates the number of X_i s fixed at the value ±1 }

GOTO / 1.

STEP 2 : (Optimality Check)

IF (*ABS* ($\sum_{i=1}^{n} F_i X_i$) ≤ GB) *AND* (ONE ≥ 1) *THEN STOP*

ELSE GOTO / 2.

STEP 3 : (Construct next solution candidate)

IF $\sum_{i=1}^{LEVEL} |X_i|$ = GOAL *THEN GOTO* / 2c.

STEP 2a : (Forward step)

IF LEVEL = n then $X_{LEVEL} := - X_{LEVEL}$ *GOTO* / 1

ELSE $X_{LEVEL} := X_{LEVEL} + 1$

MODIFY ONE

LEVEL := LEVEL + 1

$$X_{LEVEL} := \left(\sum_{i=1}^{LEVEL-1} |X_i| \right) - GOAL$$

MODIFY ONE

GOTO / 2b.

STEP 2b : (Implicit enumeration tests)

$$IF \left(\sum_{i=1}^{LEVEL-1} F_i X_i \right) - F_{LEVEL} X_{LEVEL} < -GB$$

OR

$$\left(\sum_{i=1}^{LEVEL-1} F_i X_i \right) + F_{LEVEL} X_{LEVEL} > GB$$

THEN GOTO / 2c

ELSE GOTO / 1.

STEP 2c : (Backward step)

(i) *IF* LEVEL = 1 *THEN GOTO* / 3

ELSE $X_{LEVEL} := 0$

MODIFY ONE

LEVEL := LEVEL - 1

$X_{LEVEL} := X_{LEVEL} + 1.$

MODIFY ONE

(ii) *IF* $\sum_{i=1}^{LEVEL} |X_i| = GOAL$

THEN GOTO / 2c (i)

ELSE LEVEL := LEVEL + 1

$$X_{LEVEL} := \sum_{i=1}^{LEVEL-1} |X_i| - GOAL$$

MODIFY ONE

GOTO / 2b.

STEP 3 : (No optimal solution with Q_0 = GOAL)

GOAL := GOAL + 1

IF GOAL $\leq$ Q *THEN GOTO* / 1.

CHAPTER IV

IMPLEMENTATION AND RESULTS

The solution methodologies, which were outlined in Chapter III, for the various sub-problems of the Frequency Planning Problem (FPP) are used to develop computer programs which will facilitate quick solutions. In this chapter, the organization of the various computer programs is explained along with the functional aspects of each program. The data handling done by the various programs, using disk file structures, is also explained for the purposes of understanding the input requirements and interpreting the output results. This is followed by an example of frequency planning using the system of computer programs and evaluation and interpretation of the results obtained. Directions for further work and improvements are explored as a part of the conclusion.

4.1 Overview of the Frequency Planning System

The process of frequency planning, as envisaged in this thesis, is depicted in the flow diagram in Figure 4.1. Three data files are assumed to exist at the beginning of the process - SITE.DAT, GR.DAT and COL.DAT. Information regarding incidence of the links at various sites is maintained in SITE.DAT. The graph

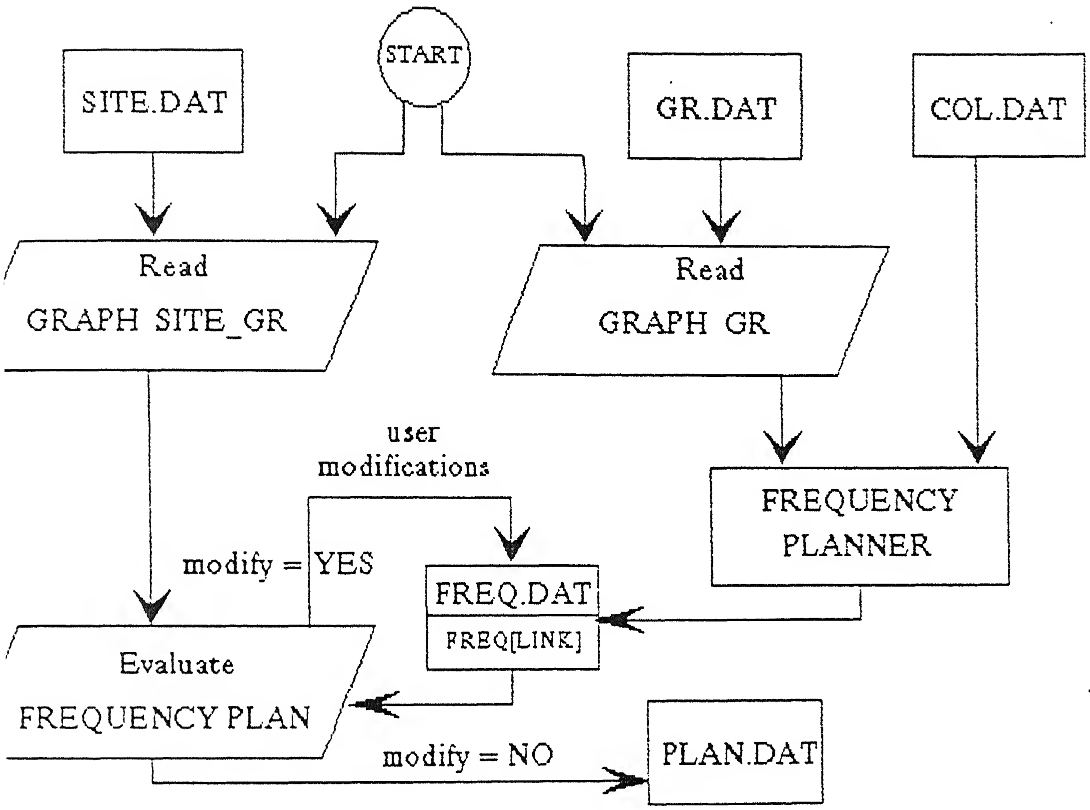


FIGURE 4.1 Flow diagram showing the frequency planning process

to be colored - GC - is stored in an adjacency list format in GR.DAT. The data file COL.DAT stores the one-to-one mapping of colors onto the set of frequencies available for use by a communication link. The planning process commences by reading into the memory SITE_GR from SITE.DAT and the graph GC from GR.DAT, both the data structures being arrays of singly linked lists as defined in PASCAL. Next, the FREQUENCY PLANNER module utilizes a graph coloring strategy to color the graph GC. The coloring strategy could be to partition the graph GC, color all the components and merge the colorings of the components to get a coloring of GC. Alternatively, the entire graph GC could be colored by a heuristic coloring algorithm. The frequency assignment for the various links is stored in FREQ.DAT. In the next stage, the user-interactive FREQUENCY PLAN EVALUATION module is used iteratively to finalize a frequency plan. In this module, the given frequency assignment is evaluated for occurrence of a significant order of intermodulation at any site in SITE_GR and based on the results, subsequent user-modification of frequency assignment to the links is permitted. A frequency plan modified by the user must undergo the evaluation described above. When no modification remains to be done, the frequency plan is stored in PLAN.DAT.

4.2 Organization of Computer Programs and Files

The computer programs, for implementing the solution methodologies for the various sub-problems of FPP, have been

coded in PASCAL using the software package Turbo-Pascal 5.0 and can be run on an IBM-compatible PC. The facility for defining units has also been exploited to enhance the modular and structured approach to programming, in general. A similar facility exists in HP-Pascal for defining modules and has also been utilized to enable the programs to run on HP super-mini computer systems.

There are two unit files which contain the various commonly used procedures. These are the unit LINK.PAS and the unit FP.PAS. The unit LINK.PAS contains the declarations regarding the data structures common across all the procedures, notably that of the *GRAPH data structure*. It also contains a *procedure R_GRAPH* which reads the formatted input from a text file and stores it in a *GRAPH*. This is nothing but the input undirected network in adjacency list format.

The unit FP.PAS has comparatively more number of procedures. These are :-

(i) *procedure MAX_FLOW* : this solves the maximum flow problem on the specified network *WGR*, between the nodes *SOURCE* and *SINK* and gets the maximum flow value as well as the minimum cut partition of the nodes of *WGR*. There are four sub-procedures contained in *MAX_FLOW*. *BFS_DIST_LABEL* performs a breadth-first-search of the working network *WGR* starting from node *SINK* to obtain

distance labels in the array *DIST*. The *sub-procedure* *ADMISSIBLE_ARC* finds out an arc, if it exists, from the present node which is admissible in the augmenting path from *SOURCE* to *SINK*. *RETREAT* backtracks from the current node if none of the arcs incident on it is admissible. *AUGMENT* increases the flow along the augmenting path identified between *SOURCE* and *SINK*.

(ii) *procedure CUT_TREE* : uses the *procedure MAX_FLOW* to develop a partial Gomory-Hu cut-tree of specified edge connectivity *K*. The cut-tree is stored in a *GRAPH* called *C_TREE*, in adjacency list format. There are many sub-procedures in *CUT_TREE*. *INIT_C_TREE* initializes the adjacency list format of *C_TREE* into a single bignode. The *SOURCE-SINK* node pair is chosen by *SELECT_SOURCE_SINK* from the same component (bignode) of *C_TREE*. *INIT_WGR* initializes the network *WGR* into a null network. *CONDENSE* carries out the process of condensing all nodes in one component of the input graph *GC* as indicated by the cut-tree *C_TREE* and constructs a simplified network *WGR*. *MODIFY_CUT_TREE* modifies the *C_TREE* after the solution of the maximum flow problem between *SOURCE* and *SINK* yields an acceptable partition of the nodes. *WRITE_GRAPH* writes the *C_TREE* developed at any stage to the output file in adjacency list format.

(iii)*procedure GR_COL* : colors the input graph using the sequential coloring algorithm for SL as well as LF ordering of the vertices. It has two sub-procedures called *ORDERING* and *SEQCOLORING*. *ORDERING* generates a SL/LF ordering of the vertices while *SEQCOLORING* colors the vertices sequentially in the order given by *ORDERING*.

(iv) *procedure WR_GRAPH* : writes the colored graph to the pre-defined output file.

(v) *procedure COL_FREQ* : reads the frequencies corresponding to the colors used by the *procedure GR_COL*, from the file COLOR.DAT.

(vi) *procedure WR_FREQ_PLAN* : writes the current frequency plan to the file FREQ.DAT.

The program COLOR.PAS uses the procedures *R_GRAPH*, *GR_COL*, *COL_FREQ* and *WR_FREQ_PLAN* to generate the *first frequency plan* in the file FREQ.DAT. As there are two possible orderings, SL and LF, of the vertices in *GR_COL*, two first frequency plans can be generated. The program EVAL.PAS evaluates the current frequency plan for critical cases of intermodulation interference and allows the user to modify it in an interactive manner to reduce the order of intermodulation. It uses the following procedures :-

(i) *procedure READ_SITE_GR* : reads the communication network as a link-site incidence list from SITE.DAT into *SITE_GR*.

(ii) *procedure READ_LINK_FR* : reads link frequencies from FREQ.DAT which is the first frequency plan, into array *LINK_FREQ*.

(iii) *procedure EVAL_INTERMOD* : evaluates the frequency assignment by checking for intermodulation of order less than *MAXORDER* at each site in *SITE_GR*. A dual backtrack algorithm is used for this purpose.

(iv) *procedure EVAL_SEPARATION* : releases a list of frequencies in an array *AV_FREQ*, which can be used for assignment to a specified link without violating the co-channel or adjacent-channel constraints. This list is used to prompt the user in modifying the frequency plan.

(v) *procedure FP_MODIFY* : allows the user to change the frequency assignment to the specified link.

The program PARTCOL.PAS uses the *procedure CUT_TREE* to generate a partition of the input graph *GC*. Next, it invokes the sub-procedures in *GR_COL* to color each partition individually.

Then *procedure* *REV_COL* is applied which, given two colored sub-graphs *GR_A* and *GR_B*, and a list *L* of edges which join a vertex in *GR_A* to a vertex in *GR_B*, gives a revised coloring of the vertices of the sub-graph formed by the merger of *GR_A* and *GR_B*. Repeated applications of *REV_COL* are used to completely color *GC* and thus produce an *alternative first frequency plan*. This frequency plan is written into FREQ.DAT and subjected to the frequency assignment evaluation program EVAL.PAS.

The listing of files is as given below :-

PROGRAM FILES	DATA FILES
LINK.PAS	COLOR.DAT
FP.PAS	SITE.DAT
COLOR.PAS	GR.DAT
PARTCOL.PAS	FREQ.DAT
EVAL.PAS	PLAN.DAT
	REPO.DAT

4.3 A Solved Example using the Frequency Planning System

The system of computer programs described earlier has been used to develop a frequency plan for a portion of the Indian Railways MW network. Table 4.1 shows the contents of the file COLOR.DAT. These are the frequency resources onto which the colors map uniquely. The portion of the IR MW network taken up for frequency planning is shown in Figure 4.2. The data for the

TABLE 4.1 Frequency Resources available for use

Color No.	Corresponding Microwave Frequency (MHz) Lower Band	Upper Band
1	7128	7289
2	7135	7296
3	7142	7303
4	7149	7310
5	7156	7317
6	7163	7324
7	7170	7331
8	7177	7338
9	7184	7345
10	7191	7352
11	7198	7359
12	7205	7366
13	7212	7373
14	7219	7380
15	7226	7387
16	7233	7394
17	7240	7401
18	7247	7408
19	7254	7415
20	7261	7422

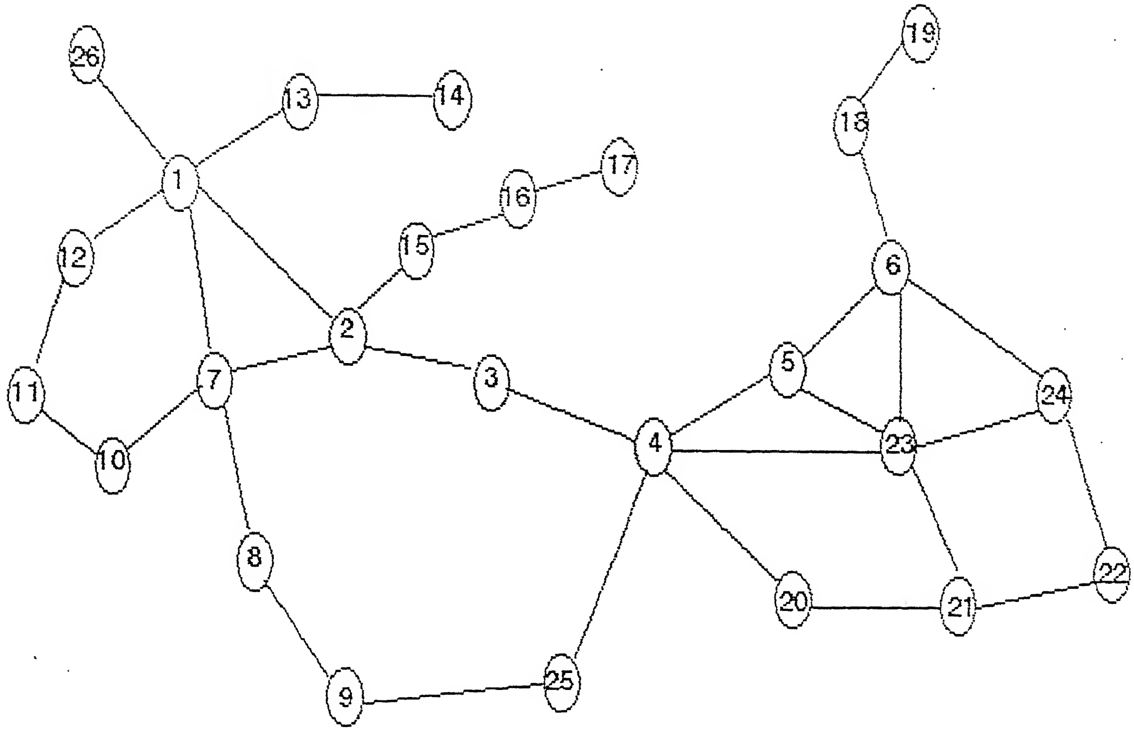


FIGURE 4.2 The example network stored in SITE_GR

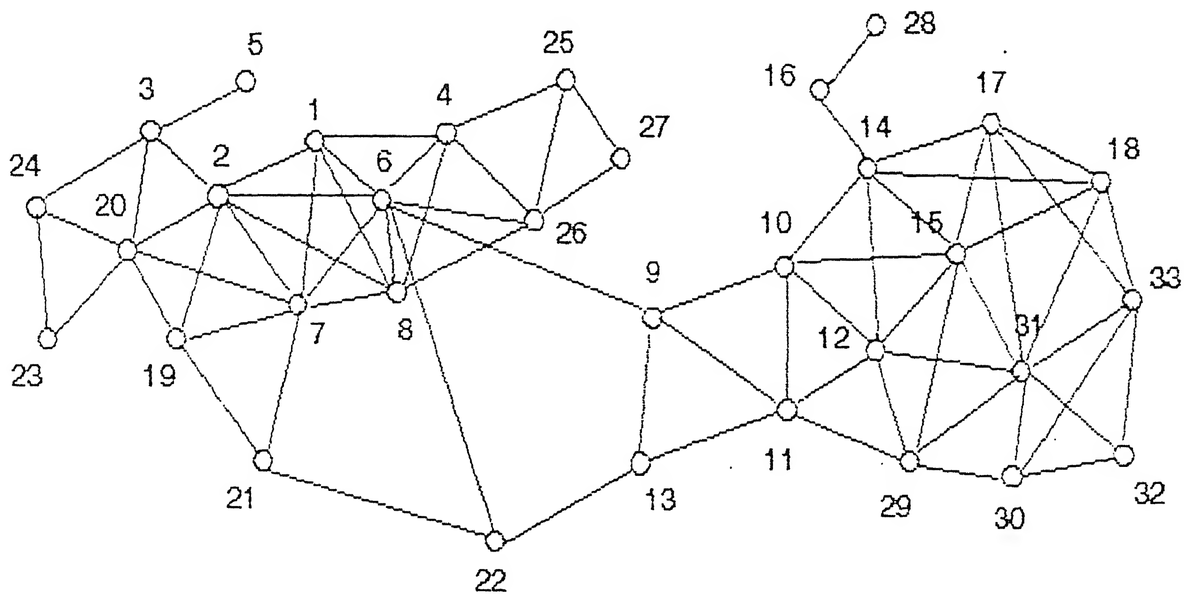


FIGURE 4.3 The graph GC to be colored

TABLE 4.2 The link-site incidence representation of the network of Figure 4.1

Site No.	Link No.s incident at this site
1	1 2 3 4 5
2	1 6 7 8
3	6 9
4	9 10 11 12 13
5	10 14 15
6	14 16 17 18
7	2 7 19 20
8	19 21
9	21 22
10	20 23
11	23 24
12	3 24
13	4 25
14	25
15	8 26
16	26 27
17	27
18	16 28
19	28
20	11 29
22	30 32
23	12 15 17 31 33
24	18 32 33
25	13 22
26	5

TABLE 4.3 Adjacency list representation of graph GC to be colored

Link No.	Adjacent Links *i.e.* interfering links
1	2 4 6 7 8
2	1 3 6 7 8 19 20
3	2 5 20 24
4	1 6 8 25 26
5	3
6	1 2 4 7 8 9 13 26
7	1 2 6 8 19 20 21
8	1 2 4 6 7 26
9	6 10 11 13
10	9 11 12 14 15 29
11	9 10 12 13 29
12	10 11 14 15 29 31
13	6 9 11 22
14	10 12 15 16 17 18
15	10 12 14 17 18 29 31
16	14 28
17	14 15 18 31 33
18	14 15 17 31 33
19	2 7 20 21
20	2 3 7 19 23 24

TABLE 4.3 contd.

21	7 19 22
22	13 21
23	20 24
24	3 20 23
25	4 26 27
26	4 6 8 25 27
27	25 26
28	16
29	10 11 12 15 30 31
30	29 31 32 33
31	12 15 17 18 29 30 32 33
32	30 31 33
33	17 18 30 31 32

TABLE 4.4 The First Frequency Plan (LF ordering)

Link No.	Frequency (MHz)
1	7191
2	7163
3	7149
4	7163
5	7128
6	7135
7	7149
8	7177
9	7163
10	7184
11	7135
12	7170
13	7149
14	7156
15	7142
16	7135
17	7184
18	7170
19	7177
20	7135

TABLE 4.4 contd.

21	7135
22	7163
23	7149
24	7163
25	7135
26	7149
27	7163
28	7149
29	7156
30	7170
31	7128
32	7156
33	7142

TABLE 4.5 A Report on Intermodulation Evaluation of Frequency Plan given in Table 4.4

SITE # 1 Order of intermodulation = 2

FREQUENCIES IN INTERMODULATION : 7163 7163
MULTIPLIER X[J] : -1 1

LINK NO. : 4 2

No. of Iterations for solution = 144

SITE # 2 Order of intermodulation is high !

SITE # 3 Order of intermodulation is high !

SITE # 4 Order of intermodulation = 3

FREQUENCIES IN INTERMODULATION : 7149 7170 7184
7163
MULTIPLIER X[J] : 1 -1 1
-1

LINK NO. : 13 12 10
9

No. of Iterations for solution = 631

SITE # 5 Order of intermodulation is high !

SITE # 6 Order of intermodulation is high !

SITE # 7 Order of intermodulation = 3

FREQUENCIES IN INTERMODULATION : 7135 7149 7163
MULTIPLIER X[J] : -1 2 -1

LINK NO. : 20 7 2

No. of Iterations for solution = 227

SITE # 8 Order of intermodulation is high !

SITE # 9 Order of intermodulation is high !

SITE # 10 Order of intermodulation is high !

SITE # 11 Order of intermodulation is high !

SITE # 12 Order of intermodulation is high !

SITE # 13 Order of intermodulation is high !

SITE # 14 Order of intermodulation is high !

SITE # 15 Order of intermodulation is high !

SITE # 16 Order of intermodulation is high !

SITE # 17 Order of intermodulation is high !

SITE # 18 Order of intermodulation is high !

SITE # 19 Order of intermodulation is high !

SITE # 20 Order of intermodulation is high !

SITE # 21 Order of intermodulation is high !

SITE # 22 Order of intermodulation is high !

SITE # 23 Order of intermodulation = 2

FREQUENCIES IN INTERMODULATION	:	7142	7142
MULTIPLIER X[J]	:	-1	1
LINK NO.	:	33	15

No. of Iterations for solution = 87

SITE # 24 Order of intermodulation = 3

FREQUENCIES IN INTERMODULATION	:	7142	7156	7170
MULTIPLIER X[J]	:	-1	2	-1
LINK NO.	:	33	32	18

No. of Iterations for solution = 143

SITE # 25 Order of intermodulation is high !

SITE # 26 Order of intermodulation is high !

TABLE 4.6 The Final Frequency Plan (LF ordering)

(after 23 iterations)

Link No.	Frequency (MHz)
1	7163
2	7205
3	7177
4	7212
5	7156
6	7128
7	7149
8	7177
9	7163
10	7184

TABLE 4.6 contd.

11	7135
12	7170
13	7177
14	7156
15	7135
16	7135
17	7184
18	7170
19	7177
20	7135
21	7135
22	7163
23	7149
24	7163
25	7135
26	7149
27	7163
28	7149
29	7156
30	7170
31	7226
32	7156
33	7198

equivalent SITE_GR (stored in SITE.DAT) is given in Table 4.2. Based on interference calculations, the graph GC to be colored is drawn up in Figure 4.3 and it's adjacency list representation (stored in GR.DAT) is given in Table 4.3.

The first frequency plans produced (for LF ordering of links) is displayed in Table 4.4. A report on the intermodulation evaluation appears in Table 4.5. The final frequency plan, produced after 23 number of iterations of user-modifications, is presented in Table 4.6.

4.4 Conclusion

The present thesis modelled the problem of frequency planning as a graph coloring problem. To color the input graph, the standard sequential algorithm is used which is an approximate one. The comparative performances of graph coloring algorithms have been evaluated in literature ([26], [33]). Consequently, a better algorithm like the RLF [33] may be implemented to generate a better first frequency plan. In fact, one of the advantages of modelling the FPP as a graph coloring problem is that any coloring algorithm developed is directly usable for solving the FPP.

The alternative strategy of partitioning the input graph, coloring the partitions and merging them, involves the development of a partial cut-tree. The maximum flow problem

solved therein used the Shortest Augmenting Path Algorithm. Since the arc capacities are generally small numbers in this case, the time spent on finding the shortest augmenting path may be saved by using any augmenting path instead. The extra time spent on the increased number of flow augmentations may be offset by the savings. In addition, other partitioning techniques like those based on removal of vertex sets or cliques, may be taken up for consideration.

The intermodulation evaluation is carried out iteratively by the user to modify the first frequency plan and eliminate intermodulation interference. This is because all the possible frequency assignments have to be implicitly or explicitly subject to intermodulation evaluation. The evaluation problem is itself a nonlinear integer program and it's solution incurs considerable computational cost. Hence, judicious intervention on the part of the user helps in modifying the first frequency plan in a small number of iterations. A method of ordering the links and frequencies may be sought before generating the first frequency plan such that the number of cases of intermodulation violation is minimal. It may be noted that the sequential coloring algorithms do order the links in a particular fashion (SL/LF) before coloring them.

A branch and bound procedure for generating an acceptable frequency plan is feasible for small networks and it may be tried out. Such a scheme works by taking a forward step or

a backward step at each stage. At any stage there exists a partial assignment of frequencies to some of the links. In a forward step, one more link is assigned a frequency; while in a backward step, the previous assignment is revoked. After each forward step, the partial assignment is checked for the interference level in each link (Formulae, like those given in Appendix I, will be useful). An upper bound on the interference level is provided by assuming that all the unassigned links have the same frequency (maximum possible interference). After each backward step, all the infeasible partial assignments are pruned as they can never lead to a feasible frequency plan. The ordering of links and frequencies can be very crucial to the running time of such a procedure.

LIST OF REFERENCES

[1] *Zoellner, J. A. and C. L. Beall*, "A Breakthrough in Spectrum Conserving Frequency Assignment Technology," *IEEE Trans. Electomagnetic Compatibility*, EMC-19, 3 (August 1977), 313-319.

[2] *Freeman, E. R.*, "An approach to Frequency Assignment," *IEEE Trans. Electomagnetic Compatibility*, EMC-8, 2 (June 1966), 90-96.

[3] *Sachs, H.M.*, "FABRIC—An Analysis Concept for Making Communication Systems Frequency Assignments," *IEEE Trans. Electromagnetic Compatibility*, EMC-9, 2 (September 1967), 78-83.

[4] *Zoellner, J. A.*, "Frequency Assignment Games and Strategies," *IEEE Trans. Electromagnetic Compatibility*, EMC-15, 4 (November 1973), 191-196.

[5] *Massaro, M. J.*, "A Multivariate Search Approach to Frequency Assignment," *IEEE Electromagnetic Compatibility Sympos.*, 1976, 47-52.

[6] *Cameron, S. H.*, "The Solution of the Graph-Colouring Problem as a Set Covering Problem," *IEEE Trans. Electromagnetic Compatibility*, EMC-19, 3 (August 1977), 320-322.

[7] *Thuve, H.*, "Frequency Planning as a Set Partitioning Problem," *European J. Oper. Res.*, 6, 1 (January 1981), 29-37.

[8] *Mathur, K., H. M. Salkin, K. Nishimura and S. Morito*, "Applications of Integer Programming in Radio Frequency Management," *Management Science*, 31, 7 (July 1985), 829-839.

[9] *Mathur, K., H. M. Salkin, K. Nishimura and S. Morito*, "The Design of an Interactive Computer Software System for the Frequency-Assignment Problem," *IEEE Trans. Electromagnetic Compatibility*, EMC-26, 4 (November 1984), 207-212.

[10] *Box, F.*, "A Heuristic Technique for Assigning Frequencies to Mobile Radio Nets," *IEEE Trans. Vehicular Technology*, VT-27, 2 (May 1978), 57-64.

[11] *Morito, S. and H. M. Salkin*, "A Survey of the Applications of Quantitative Techniques in Radio Communication Interference Problems," Technical Memorandum No. 438, Department of Operations Research, Case Western Reserve University, May 1978.

[12] *Hafez, H. M., S. Towaij and R. A. Goubran*, "The Effect of User's Geographical Distribution on the Utilization of Radio Spectrum," *IEEE Trans. Electromagnetic Compatibility*, 32, 4 (November 1990), 284-291.

[13] Approach Papers for Joint Working Group of Indian Railways and I.I.T.,Kanpur — Project for Development of Frequency Planning Software for Indian Railways.

[14] *Hansen, P.*, "Methods of Nonlinear 0-1 Programming", Paper.

[15] *Glover, F. and R. E. D. Woolsey*, "Further Reduction of Zero-One Polynomial Programs to Zero-One Linear Programming Problems," *Operations Research*, 21, 1973, 156-161.

[16] *Cooper, M. W.*, "Survey of Methods for Pure Nonlinear Integer Programming," *Management Science*, 27, 3 (March 1981), 353-361.

[17] *Appel, K. and W. Haken*, "The Solution to the Four Color Problem," *Scientific American*, 27 (4), 108-121.

[18] *Garey, M. R. and D. S. Johnson*, "Computers and Intractability: A Guide to the Theory of NP-Completeness," W. H. Freeman, San Francisco, 1979.

[19] *Aho, A. V., J. E. Hopcroft and J. D. Ullman*, "The Design and Analysis of Computer Algorithms," Addison-Wesley, Reading, MA, 1974, 364-404.

[20] *Karp, R. M.*, "Reducibility among combinatorial problems," in Complexity of Computer Computations, Plenum, NY, 1972.

[21] *Morito, S., H. M. Salkin and D. Williams*, "Two Backtrack Algorithms for the Radio Frequency Intermodulation Problem," *Appl. Math. Optim.*, 6 (1980), 221-240.

[22] *Morito, S., H. M. Salkin and K. Mathur*, "Computational Experience with a Dual Backtrack Algorithm for Identifying Frequencies Likely to Create Intermodulation Problems," *IEEE Trans. Electromagnetic Compatibility*, EMC-23, 1 (February 1981).

[23] *Nishimura, K., H. M. Salkin and K. Mathur*, "A Dual Branch and Search Algorithm to Obtain Largest IM-Free Subset from a Set of Frequencies," Technical Memorandum, Department of Operations Research, Case Western Reserve University, Cleveland, Ohio.

[24] *Syslo, M. M., N. Deo and J. S. Kowalik*, "Discrete Optimization Algorithms," Prentice-Hall, NJ, 1983.

[25] *Christofides, N.*, "Graph Theory: An Algorithmic Approach," Academic Press, London, 1975, Chapter 4.

[26] *Matula, D. W., G. Marble and J. D. Issacson*, "Graph Coloring Algorithms," in R. C. Read (ed.), Graph Theory and Computing, Academic Press, London, 1972, 109-122.

[27] *Brelaz, D.*, "New Methods to Color the Vertices of a Graph," *Comm. ACM*, 22 (1979), 251 -256.

[28] *Johnson, D. S.*, "Worst Case Behaviour of Graph Coloring Algorithms," Proceedings of the 5th Southeast Conference on Combinatorics, Grpah Theory, and Computing (1974), 513-527.

[29] *Brown, J. R.*, "Chromatic Scheduling and the Chromatic Number

Problem," Management Science, 19, 4 (1972), 456-463.

[30] *Korman, S. M.*, "The Graph-Colouring Problem," in N. Christofides, A. Mingozzi, P. Toth, and C. Sandi (eds.), Combinatorial Optimization, Wiley, NY, 1979, 211-235.

[31] *Ford, L. R. and D. R. Fulkerson*, "Flows in Networks," Princeton University Press, NJ, 1962.

[32] *Gomory, R. E. and T. C. Hu*, "Multi-terminal Network Flows," *J. SIAM*, 19 (1961), 551-570.

BIBLIOGRAPHY

[1] *Carlson, A. B.*, "Communication Systems," McGraw-Hill.

[2] *Hu, T. C.*, "Combinatorial Algorithms," Addison-Wesley, 1982.

[3] *Minoux, M. and Gondran, M.*, "Graphs and Algorithms," Wiley Interscience, 1984.

[4] *McHugh, J. A.*, "Algorithmic Graph Theory," Prentice Hall, NJ, 1990.

[5] *Salkin, H. M. and K. Mathur*, "Integer Programming," Book.

APPENDIX I

1 A technical overview of the operation of a link in a communication network

The scheme of operation of a point-to-point link in a communication network is illustrated for the case of a *Microwave system*. (See Figure I.1) The process of a signal being relayed from Site 1 to Site 2 involves the following steps :-

(i) A *signal* to be relayed is generated by the concerned equipment.

(ii) The *carrier wave* (it has a frequency in the microwave spectrum range, which is very large compared to the signal wave frequency) is generated using a special device called oscillator which could be a vacuum tube device like the *klystron*.

(iii) The *modulator* combines the signal wave with the carrier wave and feeds the output to the transmitter after filtering out spurious emissions at frequencies outside the neighbourhood of the modulated wave frequency. The effect of filtering is to reduce the effect of adjacent channel interference (described under Section 3) but the filtering may not be adequate always.

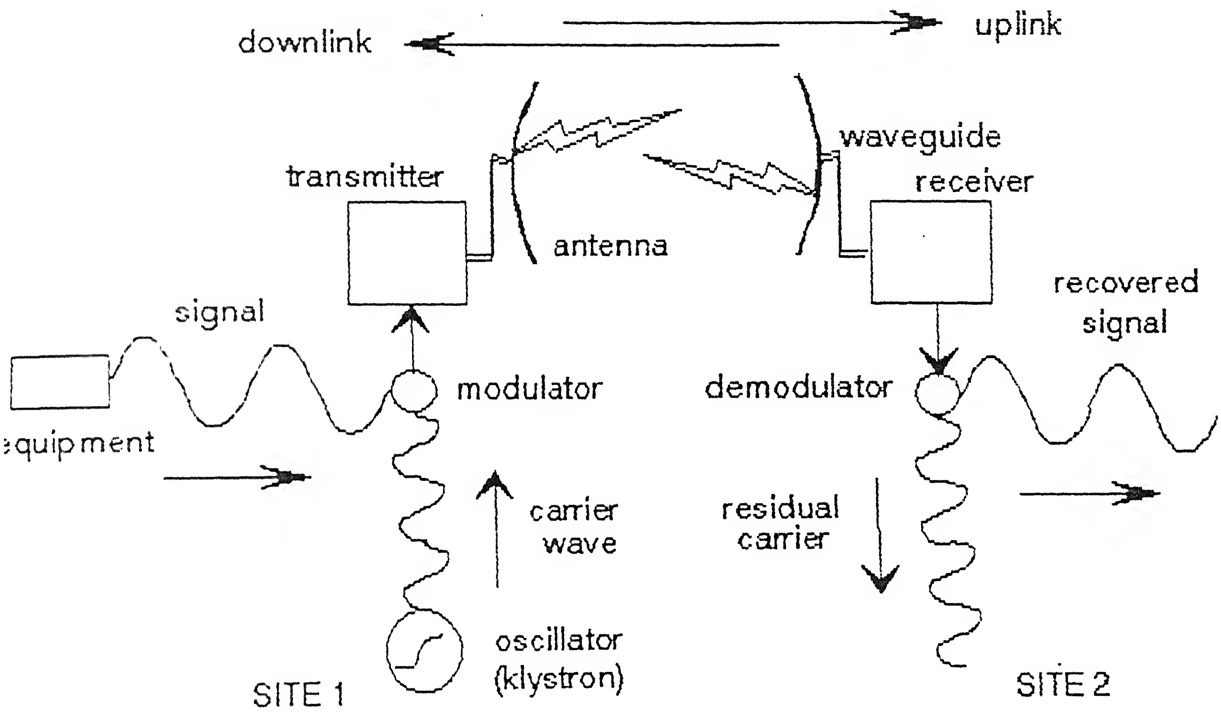


FIGURE I.1 Microwave communication link

(iv) The *transmitter* boosts the power of it's input wave (signal + carrier) upto the required levels and feeds it to the antenna via the *waveguide*.

(v) The *antenna* in turn, by the virtue of it's geometry, beams the input wave in the direction of the receiving antenna.

(vi) The receiving antenna feeds the received wave to the *demodulator* via the waveguide.

(vii) The demodulator separates the signal wave from the carrier wave using a process similar but inverse to that of modulation. It also employs filtering to remove interfering signals at unwanted frequencies.

Wherever, the text of the thesis contains a mention of frequency planning, the frequencies being referred to are actually those of the carrier waves which fall in the microwave range of the spectrum. Depending upon the type of service provided by the communication network, it can avail of the frequency bands (and channels thereof) allocated to that type of telecommunication services at the international and national levels. An additional consideration would be the previous frequency assignments made in the same frequency band, if any, for an existing network providing similar telecommunication services. So the frequency planner for the communication network ends up with a small set of discrete frequency channels out of

the entire usable spectrum. This set of frequencies constitutes the electromagnetic resources shared by all the links of the communication network.

A *simultaneous two-way communication* between the two sites on a link, called a *duplex* link, will require two different frequencies for the up-link and the down-link respectively, because the signal paths for up-link and down-link are the same and the two signals can potentially interfere with each other. A radio telephone link is an example of a duplex link. A *half-duplex* link, on the other hand, allows two-way communication on a single link frequency but simultaneous operation of the up-link and the down-link is not permissible, as for example in the case of wireless sets. The *simplex* link is a one-way communication only, using a single link frequency, as in radio broadcasting networks.

If a site is linked to more than one site, then a *common antenna and waveguide* might be used in the transmitter / receiver system to meet the requirements of some or all the links incident at this site. The *antenna geometry* is usually *parabolic* with the signal wave being fed at it's focus, either directly from a *horn* or via a *reflector*. The signal wave at the focus of the parabolic antenna gets transmitted as a parallel beam after reflection from the parabolic surface of the antenna. The *directionality* lent by the antenna to the wave beams restricts their spherical dispersion in the atmospheric media and reduces

the spillover of the (interfering) signals into 'wrong' receivers. This causes an attenuation of the signal in the directions different from the direction in which the signal wave is beamed and is called angular attenuation. In such a case, a greater re-use of frequencies may be possible and this leads to increased utilization of the frequency spectrum.

Polarization of a wave restricts the wave motion to a plane. (See Figure I.2) If two non-polarized waves, incident at a receiver, are potentially interfering with each other then cross-polarization of the two waves (the two waves are plane-polarized in orthogonal planes) will reduce the mutual interference potential. So frequency re-use can be increased by resorting to this technique.

2 Transmission Losses during the propagation of a radio wave

In any communication system, the transmitted wave suffers a loss in power as it propagates away from the transmitter and reaches the receiver. To quantify the amount of loss taking place, a few definitions are introduced below

A linear time-invariant communication system has a (power) *gain* defined as

$$g = \frac{P_{in}}{P_{out}} .$$

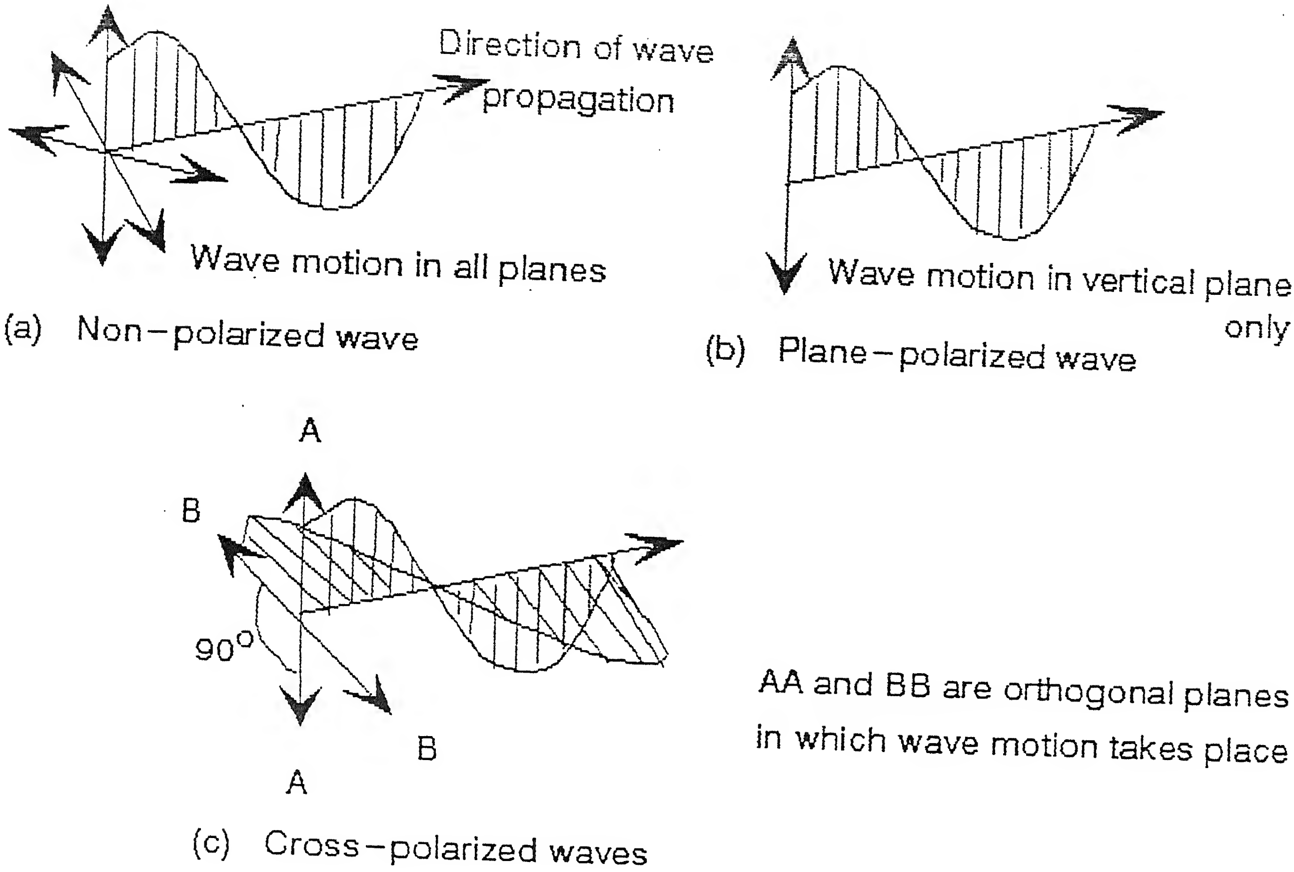


FIGURE I.2 Polarization of signal carrying waves

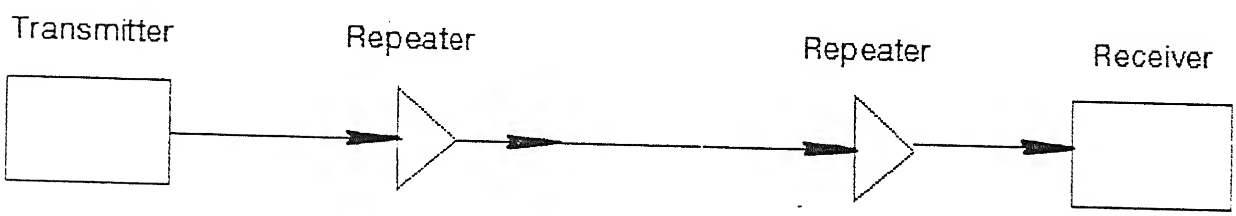


FIGURE I.3 Arrangement of repeaters in a (Cable) transmission system

Since the possible values of g may be very large, it is convenient to express gain in logarithmic units *i.e.*

$$g_{dB} = 10 \log_{10} g \, .$$

Sometimes the signal strength (P) itself is expressed in logarithmic units, such as,

$$P_{dBW} = 10 \log_{10}\left(\frac{P}{1\ W}\right) .$$

Any passive transmission system has a power loss rather than gain, since $P_{out} < P_{in}$. So it is preferred to work with transmission loss or *attenuation* defined as,

$$L = \frac{1}{g} = \frac{P_{in}}{P_{out}} \text{, and}$$

$$L_{dB} = - g_{dB} = 10 \log_{10}\left(\frac{P_{in}}{P_{out}}\right).$$

Signal transmission by cables is very lossy : a lot of repeaters (mid-link power amplifiers where the signal power along with the noise content gets amplified) are needed. (See Figure I.3) On the other hand radio propagation (Microwave is at the upper extreme of the Radiowave portion of the spectrum) reduces the required number of repeaters and eliminates the cost of cables. Furthermore, large bandwidths are available for use in the microwave range which means more information can be carried by the communication channels.

A *line-of-sight propagation* means the radio wave travels a direct path from the transmitting to the receiving antenna. The *path loss* or the *free-space loss* on line-of-sight path is due to spherical dispersion of the radio wave. This loss is given by

$$L = \left(\frac{4\Pi l}{\lambda}\right)^2 = \left(\frac{4\Pi f l}{c}\right)^2 \text{, where}$$

λ = signal wavelength;

f = signal frequency;

l = path length; and

c = speed of light.

Expressing l in km and f in GHz (range of Microwave), we get

$$L_{dB} = 92.4 + 20 \log_{10} f_{GHz} + 20 \log_{10} l_{km} .$$

Directional antennas have a focussing effect that acts like amplification , *i.e.*

$$P_{out} = \frac{g_T \; g_R}{L} P_{in} \text{, where}$$

g_T and g_R are antenna gains at the transmitter and receiver respectively and L is the path loss. The maximum transmitting or receiving gain of an antenna with effective aperture area A_e is

$$g = \frac{4\Pi A_e}{\lambda^2} = \frac{4\Pi A_e f^2}{c^2} \text{, where}$$

$c = 3 \times 10^5$ km/s.

The value for A_e for a horn or dish antenna approximately equals it's physical area, and large parabolic dishes may provide gains

in excess of 60 dB. As described under Section 1, the focussing effect of the directional antenna also implies attenuation in directions other than the antenna direction.

3 Interference Mechanisms

Interference refers to the contamination of an information bearing signal by another similar signal, usually from a human source. This occurs in radio communication when the receiving antenna picks up two or more signals in the same frequency band. Interference may also result from multi-path propagation. Regardless of the cause, severe interference prevents successful recovery of the message information.

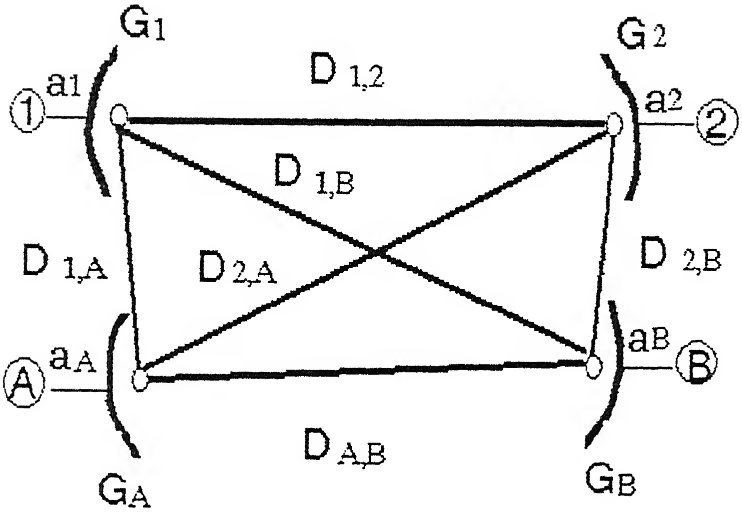
.1 Propagation model of interference

The interference model is built on a representation as shown in Figure I.4. The RF-decoupling is calculated by the formula below

$$a_{RF} = D + a_T + a_R + \Delta G + \Delta a + \delta , \qquad (I.1)$$

here

$D = 20 \log \left[\frac{D_{is}}{D_{ws}}\right]$ = difference between propagation loss in interfering and wanted signal paths;



G = antenna gain

a = attenuation between antenna and transmitter output

D = path length

$D_{1,2}$, $D_{A,B}$: Wanted signal paths

$D_{1,A}$, $D_{1,B}$, $D_{2,A}$, $D_{2,B}$: Interfering signal paths

A,B : MW stations of interfering hop

1,2 : MW stations of sample hop

FIGURE I.4 Propagation model of interference

a_T = angle attenuation at interfering transmitter;

a_R = angle attenuation at interfered receiver;

ΔG = difference between antenna gains at wanted and interfering signal transmitters;

Δa = difference between attenuations at wanted and interfering signal transmitters;

δ = additional obstructions in interfering signal path.

The above model of interference is based on the propagation formulae of signal waves. It does not characterize the type of interference taking place and also does not know whether a particular type of interference will actually take place. Such a model can be referred to as the *propagation model of interference*.

There are three types of interference possible *i.e.*

(i) Co-channel interference

(ii) Adjacent-channel interference

(iii) Receiver/Transmitter Intermodulation interference.

Interference calculations, based on the propagation model, must therefore be made for each of the above possibilities.

.2 Co-channel model of interference

Both the desired and the undesired signals are carried in the same frequency channel and the receiver is unable to

A = that amount in dB by which the side-band noise is below the carrier power;

B and C are constants; and

ΔF = frequency offset from the carrier wave frequency at which the sideband noise strength is to be estimated.

Such an equation has to be constructed for each carrier wave frequency range. Obviously, the value for A depends upon the efficacy of filtering carried out at the transmitter. Adjacent-channel interference exists if the value of A is smaller than a specified threshold value. The extent of adjacent-channel interference can also be calculated by calculating the amount of interference based on the the propagation model and then applying to it the Interference Reduction Factor (IRF) which depends upon the frequency offset (ΔF).

3.4 Receiver/Transmitter Intermodulation interference model

This can occur at both the transmitter and the receiver and is a result of nonlinear behaviour of the circuitry at the transmitter output/receiver input stages. Interactions occur between the various signals present at a site because the circuitry is responsive to all of these signals, though not equally so.

Receiver intermodulation occurs when a receiver frequency is expressible as an integer linear sum of the interfering signal frequencies (*i.e.* coefficients of the linear

terms in the sum are all integers). The sum is also called the intermodulation product frequency and the sum of the absolute values of the integer coefficients is called the order of the intermodulation product. The amplitude of the product wave depends upon it's order as well as it's frequency offset from the victim receiver's frequency. Increased linearity in the circuitry at the receiver input stage helps in reducing the intermodulation interference.

Transmitter intermodulation occurs when a signal received from one transmitter gets combined with the output signal of a second transmitter in it's output stage to produce intermodulation product frequencies which are radiated by the second transmitter. Selective filters/isolators at the output stage of the second transmitter is useful for reducing the amplitude of the signal received from the first transmitter. As the path loss of the involved signals is double that for a receiver intermodulation, the transmitter intermodulation products are considerably weaker in amplitude.

Intermodulation interference is said to occur if the amplitude of the intermodulation product is greater than the specified threshold value at the victim receiver.

4 International/National Regulations for Frequency Usage

International agencies like CCIR and ITU (International

Telecommunication Union) whose membership comprises of almost all the nations of the world, administer the usage of electromagnetic spectrum. The world has been divided into various regions for the purposes of regulating frequency usage. Each region has an international allocation of frequency bands/channels for use in a specified type of telecommunication service. The national allocations are also made similarly for the distribution of spectrum resources amongst the various users. The frequency plan prepared by each user (for 'his' network) must be registered with the National Frequency Registration Board for proper documentation and administration of the national frequency allocations.

Keeping in view the above issues the frequency planner should be aware of all the relevant regulations governing frequency usage in 'his' geographical zone and also in the type of telecommunication service 'his' networks intends to provide.

APPENDIX II

1 The Indian Railways Microwave Network

The Indian Railways has got it's own microwave network, spanning the entire nation, to meet it's communication needs. There are about 100 base stations (transmitter/receiver sites) and about 300 hops (communication links) spread almost uniformly all over India. Presently, the network is being modernized from operating analog systems to using digital microwave systems.

As per the plan prepared by the Indian Railways, there will be three distinct types of networks during three different time periods :-

(i) Existing Analog Network - The communication links are designed to handle only analog signals. This network will be in use till the implementation of the digital microwave systems.

(ii) Planned Digital Network - New digital systems will be deployed along with the retained analog systems for the microwave links.

(iii)Future Digital Network - This will be predominantly digital with some analog systems retained.

From the above description it is clear that there is a

transition from using analog systems to operating digital systems and the transitional phase will witness analog and digital systems co-existing in the planned digital network . This is because the analog systems cannot be shut down abruptly to deploy the digital systems - care has to be taken that the links are not shutdown, as the telecommunication needs of the Indian Railways are of a continuous nature.

2 Objectives of Frequency Planning

The objectives pursued by Indian Railways for the frequency planning of it's microwave network are long-term in nature and they are broadly three-fold :-

(i) There should be a stable, dominantly digital microwave network to meet the Indian Railways telecommunication needs.

(ii) The transition from the existing network operations to the future network operations must be smooth, especially during the interim planned digital network phase when digital and analog systems must co-exist. The frequency plan for the planned network should be such that little alteration is required in the future network and the shutdown cases for the analog systems in the planned digital network are kept to the minimum.

(iii) Frequency planning for the planned digital network is most crucial and must be carried out after taking into consideration the requirements of the future network.

3 Process chart for Manual Frequency Planning

The process chart in Figure II.1 illustrates the methodology followed for manual frequency planning. The Survey Map shows all the base stations with their coordinates (*i.e.* latitude and longitude) while the Sample Hop Data pertains to the technical details of the hop under study for interference from other hops within it's coordination area. The Concept is based upon the formulae for compatibility calculations and it also gives guidelines for frequency allocations based on mandatory regulations and experience. It also stipulates the minimum requirement of RF-decoupling for each link in order to avoid interference from other links.

The data of the sample hop is marked on the survey map for ready reference. A Route Map is drawn, showing the various interfering hops (within the coordination area around the sample hop) for which compatibility must be established.

The frequency arrangements are made for the sample hop with the polarization direction also being determined. The frequency channels are chosen such that compatibility calculations are most likely to succeed.

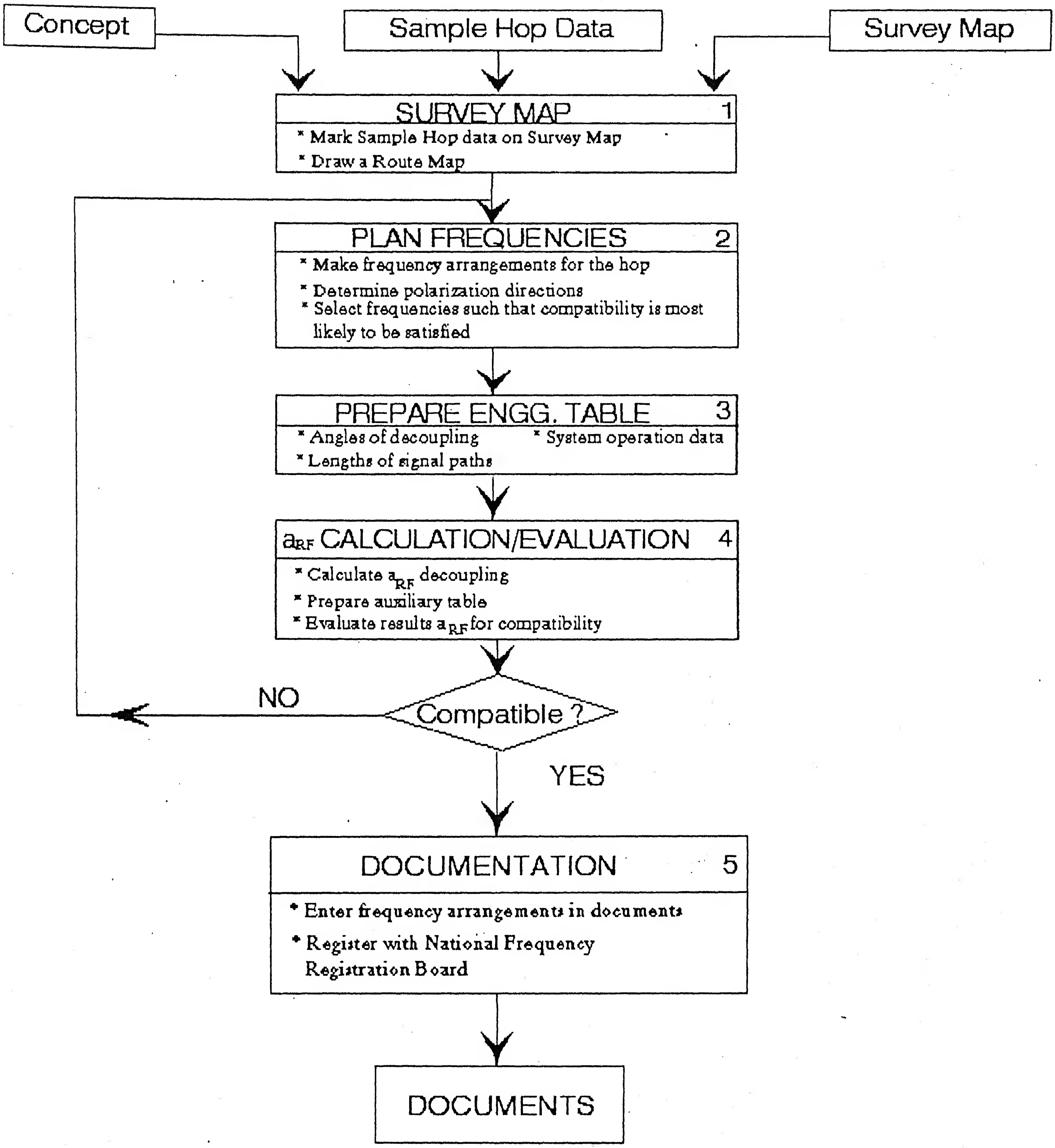


FIGURE II.1 Manual Process of frequency planning

An Engineering table is prepared for ready reference while making compatibility calculations. Besides the system operation data, angles of RF-decoupling and lengths of the wanted and the interfering signal paths are also included in it.

For ease of evaluating the RF-decoupling value (a_{RF}) an auxiliary table is prepared using which the a_{RF} value is calculated and the results are evaluated to check if the compatibility is satisfied as per the concept.

If the compatibility is not established then the iteration is repeated afresh by changing the frequency arrangements. Once compatibility is secured, the frequency arrangement is documented and registered with the National Frequency Registration Board for the purposes of information and administration.

4 Computerized Frequency Planning Process

The computerization of the frequency planning process, as outlined in the Indian Railways report [*], is to be carried out using a mainframe computer. It is for most parts an automation of the existing manual process of frequency planning. It involves creation and maintenance of large databases for the system models, the base stations, and the hops. The various fields of data stored in each of these databases are as given

below :-

(i) Model - Microwave System used (FM 120/FM 300/FM 960/DRS 34)

- Antenna Type (Standard Performance/High Performance)
- Compatibility requirements

(ii) Station - Altitude of site

- Tower height
- Coordinates of site

(iii)Hop - System

- Antenna height
- Antenna feeder attenuation
- Power attenuation
- Frequency arrangement
- Polarization
- Number of frequency channels required
- Distinct network (Existing/Planned/Future)
- Data of frequency channels

Some of the field values must be filled in two records for each hop - one for each end station.

System operation data is available for each microwave system including information regarding each of the following :-

- Co-channel decoupling required
- Minimum angular separation required for each antenna type
- Polarization permissible

- Frequency range and spacing
- Attenuation at various stages in the link
- IRF curves

The compatibility calculations are made by a computer program called the HOP-ENGINEERING routine. It is called by specifying the Hop keys (two keys - one for each station) and the System key. The calculations are made using a compatibility matrix which is already calculated and stored. The output is a listing of those critical cases where compatibility is not satisfied. The database updation is carried out using a separate computer program called the HOP-DIALOGUE routine.

Discussion during the Thesis-Examination

1. Since the intermodulation evaluation module consumes a lot of time in iterations and it's effectiveness is also dependent upon the skills of the user, it was felt that subsets of intermodulation-free frequencies should be generated prior to entering the coloring module by using the same nonlinear programming technique. Alternatively, a heuristic suggested was to group frequencies such that the ratios between any two frequencies are certain types of numbers (for example, 0.15, 0.85, 0.7 etc.). This might result in the group of frequencies to be intermodulation-free.

2. An interesting observation was that instead of partitioning the graph GC to be colored using the property of minimum cut, one could try to see if some "nicely-colorable" subgraphs can be obtained by deleting some vertices of GC. For instance, if planar subgraphs are obtained, they are easily 5-colorable (in principle, they are 4-colorable). This property would be desirable if the subsets of frequencies which are intermodulation-free are of relatively small cardinality (5 to 10, for example). Once the subgraphs are colored, merging occurs by re-introducing the vertices deleted earlier with a suitable coloring, one-by-one.

The methodology of recognizing "nicely-colorable" subgraphs has to be examined further.